SYMANTEC.™

VisualCafé™
Getting Started

Symantec VisualCafé™ Getting Started

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Copyright Notice

Released: 11/99 for VisualCafé 4.0

Trademarks

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

SYMANTEC LICENSE AND WARRANTY

The software which accompanies this license (the "Software") is the property of Symantec or its licensors and is protected by copyright law. While Symantec continues to own the Software, you will have certain rights to use the Software after your acceptance of this license. Except as may be modified by a license addendum which accompanies this license, your rights and obligations with respect to the use of this Software are as follows:

• You may:

(i) use one copy of the Software on a single computer;

(ii) make one copy of the Software for archival purposes, or copy the software onto the hard disk of your computer and retain the original for archival purposes;

(iii) use the Software on a network, provided that you have a licensed copy of the Software for each computer that can access the Software over that network;

(iv) after written notice to Symantec, transfer the Software on a permanent basis to another person or entity, provided that you retain no copies of the Software and the transferee agrees to the terms of this agreement; and

(v) if a single person uses the computer on which the Software is installed at least 80% of the time, then after returning the completed product registration card which accompanies the Software, that person may also use the Software on a single home computer.

• You may not:

(i) copy the documentation which accompanies the Software;

(ii) sublicense, rent or lease any portion of the Software;

(iii) reverse engineer, decompile, disassemble, modify, translate, make any attempt to discover the source code of the Software, or create derivative works from the Software; or

(iv) use a previous version or copy of the Software after you have received a disk replacement set or an upgraded version as a replacement of the prior version, unless you donate a previous version of an upgraded version to a charity of your choice, and such charity agrees in writing that it will be the sole end user of the product, and that it will abide by the terms of this agreement. Unless you so donate a previous version of an upgraded version, upon upgrading the Software, all copies of the prior version must be destroyed.

• Sixty Day Money Back Guarantee:

If you are the original licensee of this copy of the Software and are dissatisfied with it for any reason, you may return the complete product, together with your receipt, to Symantec or an authorized dealer, postage prepaid, for a full refund at any time during the sixty day period following the delivery to you of the Software.

• Limited Warranty:

Symantec warrants that the media on which the Software is distributed will be free from defects for a period of sixty (60) days from the date of delivery of the Software to you. Your sole remedy in the event of a breach of this warranty will be that Symantec will, at its option, replace any defective media returned to Symantec within the warranty period or refund the money you paid for the Software. Symantec does not warrant that the Software will meet your requirements or that operation of the Software will be uninterrupted or that the Software will be error-free.

THE ABOVE WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE.

• Disclaimer of Damages:

REGARDLESS OF WHETHER ANY REMEDY SET FORTH HEREIN FAILS OF ITS ESSENTIAL PURPOSE, IN NO EVENT WILL SYMANTEC BE LIABLE TO YOU FOR ANY SPECIAL, CONSEQUENTIAL, INDIRECT OR SIMILAR DAMAGES, INCLUDING ANY LOST PROFITS OR LOST DATA ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF SYMANTEC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

IN NO CASE SHALL SYMANTEC'S LIABILITY EXCEED THE PURCHASE PRICE FOR THE SOFTWARE. The disclaimers and limitations set forth above will apply regardless of whether you accept the Software.

• U.S. Government Restricted Rights:

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software-Restricted Rights clause at 48 CFR 52.227-19, as applicable, Symantec Corporation, 10201 Torre Avenue, Cupertino, CA 95014.

• General:

This Agreement will be governed by the laws of the State of California. This Agreement may only be modified by a license addendum which accompanies this license or by a written document which has been signed by both you and Symantec. Should you have any questions concerning this Agreement, or if you desire to contact Symantec for any reason, please write:

Symantec Customer Service, 175 W. Broadway,

Eugene, OR 97401

Language Addendum

If the Software is a Symantec language product, then you have a royalty-free right to include object code derived from the Symantec component (java source or class) files in programs that you develop using the Software and you also have the right to use, distribute, and license such programs to third parties without payment of any further license fees, so long as a copyright notice sufficient to protect your copyright in the program is included in the graphic display of your program and on the labels affixed to the media on which your program is distributed. You have the right to make changes to the Symantec components, but only to the extent necessary to correct bugs in such components, and not for any other purpose. You also have a royalty-free right to include unmodified (except as stated in the previous sentence) Symantec component files required by your programs, but not as components of any development environment or component library you are distributing. The Symantec component files that may be redistributed are in the following folder in the VisualCafé directory - VisualCafe\redist. The Java Virtual Machine (VM) or Just In Time (JIT) compiler may not be redistributed.

C O N T E N T S

CHAPTER 1

Introducing VisualCafé

Symantec's VisualCafé Standard Edition is an integrated development and debugging environment for creating Web-hosted Java applets and standalone Java applications. Additional features such as Java component libraries, graphics libraries, and templates provide the complete solution for the Java developer or sophisticated Web developer.

With the VisualCafé Expert Edition, the same powerful environment also provides database connectivity. The VisualCafé Expert Edition is an extension to the VisualCafé environment that comprises a full suite of integrated Java database tools. These enable you to quickly create powerful, databound Java programs by using wizards that provide assistance at each step of the development process. You can also write source code directly if you like.

VisualCafé is easy enough to use for the novice user, yet powerful and extensive enough for the most experienced Java programmer.

This book, *VisualCafé Getting Started*, along with the tour files installed with VisualCafé, introduces you to the major features of VisualCafé Expert Edition.

Features of VisualCafé

VisualCafé gives you the easy-to-use Java language in a development environment designed to make you highly productive. It includes the following features, which you'll learn more about in the tour:

- Visual, form-based development
- Quick creation of component connections with the Interaction Wizard
- Two-way development so that the visual tools and Java code always match
- Database connectivity through a three-tier architecture and the JDBC standard
- Support for major database servers
- Database wizards for simplified form development
- Hierarchical view of database cataloging information with the dbNAVIGATOR
- Simplified Java database connectivity through databound components
- Support for JDK 1.1 and Java 2, portable JavaBeans components, and Java Archive (JAR) files
- Quick creation of component interactions
- Fast compilation by way of the Symantec Just-In-Time compiler.
- An advanced debugger that includes expression evaluation

About the tour

The major features of VisualCafé Expert Edition are illustrated in the tour as you build an order administration system for the fictional Hartwig Pharmaceutical Company.

This section provides an overview of the features that are used in the tour.

VisualCafé core features

The tour begins in Chapter 3, "VisualCafé Fundamentals." In this chapter you will be introduced to some of VisualCafé's fundamental tasks and procedures while you are building a customer information applet for the Hartwig Pharmaceutical Company. This activity includes the following tasks:

- Creating and working with a VisualCafé project
- Adding Swing components to an applet
- Running an applet in the VisualCafé environment
- Adding interactions to a Swing component
- Inserting sample data to your form
- Adding code to the applet source
- Including an HTML file with the Java project files
- Creating Java Archives (JARs) for the finished files
- Deploying the project
- Debugging a VisualCafé project
- Preparing a VisualCafé project for localization

The Swing components (also known as JFC) are the new set of Java components. They replace the AWT (Abstract Windowing Toolkit) components. (The AWT components are still included with VisualCafé and Java if you want to use them.) The design of the Swing components gives you more control over the appearance and behavior of your application. See the Swing chapter of the *VisualCafé User's Guide: Standard Features* for more information.

How much programming do I need to know?

In VisualCafé it's possible to develop Java programs without writing a single line of source code. There are many ways to obtain Java programs, such as books, the Internet, and your friends and colleagues. You can drop these programs into VisualCafé and have an applet, application, or Bean that's ready to use. However, sometimes these programs need

modifications, and that's where real challenges to your programming expertise occur.

This manual assumes that you're familiar with the Java programming language or are learning how to program in Java. It's beyond the scope of this manual to teach Java programming, although you can learn more about Java by using VisualCafé. To learn how to program in Java, you can consult one of the many excellent books that extensively explore the Java language. You can also search the Internet and find many well-written Java tutorials and summaries, as well as abundant resources to guide you on your way to becoming a Java developer. You might want to participate in a Java programmers' special interest group (SIG) in your area. You can find user groups on the Web or through Usenet newsgroups.

To use VisualCafé, it helps to have a basic understanding of object-oriented programming languages, such as C++. Many of the principles and concepts of Java are based on those found in C++, although you should note that there are also some vast differences between Java and C++.

You should also have a basic understanding of cross-platform operating system concepts. This knowledge is useful, for example, when you're developing multithreaded Java programs, because all platforms handle threading differently.

If you're using VisualCafé Expert Edition, you need to have basic Microsoft Windows programming skills in order to develop native 32-bit applications and libraries. The *VisualCafé User's Guide: Expert Features* shows you how to create native Win32 applications.

Also, a basic understanding of Structured Query Language (SQL) and client-server models is necessary if you want to build complex databound Java programs.

Finally, a basic understanding of Hypertext Markup Language (HTML) is helpful so you can develop relationships between your Java applets and Web pages.

Working with Java

In the tour and throughout the VisualCafé environment, you use Java to create programs called applets and applications. These programs can work

over the Internet, an intranet, and other heterogeneous, distributed networks.

The VisualCafé Expert Edition generates JDBC code to connect your project to any JDBC-compliant database. In addition, the VisualCafé Expert Edition adds some of its own classes and interfaces to the JDBC standard set, which were created by Sun Microsystems to extend the Java language. These extensions allow your application or applet to connect and communicate with a database.

Java applets and Web browsers

If you've explored the World Wide Web with a Java-enabled browser, you've probably run into Java applets such as steaming coffee cups, fireworks, hotel and airline reservation utilities, or even adventure games. Integrating an applet into a Web page is as easy as adding a graphic: you just add a line of HTML code. When you access a Web page that has an applet, the applet code is transferred to your computer and executed by the Web browser. This reduces the load on the Web server and makes the applet run faster.

You can include new types of data and provide associated Java bytecode programs that know how to process the data on the client computer. For example, if you wanted your applet to display an image that is stored in a format not supported by most browsers, you could incorporate the ability to handle this graphics format in the applet.

With VisualCafé, it's easy to build Java applets that add dynamic, interactive functionality to your Internet or intranet Web site. You just create a new project and open the applet in the Form Designer, where you design the applet with visual tools. To create your user interface, you can drag onto the form a variety of components, such as a button, a text field, and even an animation or slide show, then position or resize the components as needed. You assign the component properties in a separate Property List window and add interactions between components with the Interaction Wizard. VisualCafé generates the Java code for you during this design process. To test the applet, you can run it with VisualCafé.

After your applet is finished, you can optionally add an HTML file to the project. You can create and view HTML files using the Source window in VisualCafé, or you can create the files in another environment and add them to a VisualCafé project.

Because of VisualCafé's extensive library of components, you can develop many applets without manually creating or editing Java code. To program other features, you can manually add Java code in the Source window. For example, you can add custom data processing or write your own components then add them to the Component Library.

About Java applications in VisualCafé

Java applications are cross-platform, which makes them an ideal choice for intranets. A Java application can run on all computers that have Java, so separate applications for each environment aren't needed. Unlike Java applets, Java applications typically have menu bars and can access files on the local computer. This additional functionality lets you implement features in applications that you can't provide in an applet, but it also makes applications less secure than applets. You should be aware of this if you are going to write code to access local storage or any host other than a Web server.

Like other applications, Java applications don't need to run from a Web browser. A Java application is like a C++ application, but it's cross-platform and you need a standalone Java virtual machine and supporting files on your computer to run it.

With VisualCafé, you can quickly and easily develop a Java application using an application template that creates a main window with a menu bar containing typical menu commands. The standard application template also generates an About box and an exit confirmation dialog.

After creating the application, you can open the main window in the Form Designer. This enables you to design the Java frame with visual tools. You can create your application with the same tools that you use to create applets.

What is JDBC?

The many advantages of Java make it an excellent choice for use with databases. JDBC is an SQL-level API that provides Java developers with database access that is independent of the database and the connectivity mechanism. The generic SQL database access framework in JDBC provides a uniform interface to a variety of database connectivity modules.

Note: For an overview of database-related terms and concepts, see the *VisualCafé User's Guide: Expert Features*.

VisualCafé is fully compliant with the JDBC API, so that you can use VisualCafé to create a single database interface that's compatible with a variety of connectivity solutions. The JDBC design is similar to the Open Database Connectivity (ODBC) standard, so its interface is familiar to many programmers.

By using VisualCafé to create databound applets and applications, you generate JDBC code which enables you to:

- Connect to a database.
- Send SQL statements to query your database.
- Generate query results.

Examples of databound applets might be a corporate phone list or an e-mail form which allows you to receive feedback from a Web page. An application might be a stock reporting system, which gathers information from a database on the Web and stores it in a database on your local machine.

VisualCafé documentation

VisualCafé comes with extensive documentation and online help to assist you in the process of developing applets and applications. Besides this manual, which was described earlier, the VisualCafé documentation set is described below.

Online Help

VisualCafé has extensive online help that describes all of the procedures for building Java applets and applications. To access VisualCafé's Help, choose Help Topics from the Help menu. The online help is also context-sensitive, which means that you can press F1 in most areas of VisualCafé and receive information that pertains to your current activity.

You can also access information about VisualCafé's components and review the Java API documentation from Sun Microsystems. You can access information about an individual component by typing the name of the component in the Index tab of the Help window, choosing Components Reference from the Help window's Contents tab, or selecting a component in the Component Library and pressing F1.

VisualCafé also includes online help for the Java macro system. To access this help topic, choose Macro Reference from the Help menu.

ReadMe file

The ReadMe file is the first document you should read before using VisualCafé. It contains late-breaking news, work-arounds, and known issues. This file available for viewing at the end of the installation process, as well as from the Windows Start menu.

VisualCafé User's Guide: Standard Features

The *VisualCafé User's Guide: Standard Features* may be the document you turn to most frequently as you work with VisualCafé. It contains both conceptual information and step-by-step procedures. It describes VisualCafé's development environment and covers the basics of project development, with step-by-step instructions for creating, compiling, running, and debugging a project.

VisualCafé User's Guide: Expert Features

The *VisualCafé User's Guide: Expert Features* describes the additional features that are included in the Expert Edition of VisualCafé, such as database access tools, Symantec's high-performance compiler, and one-step deployment. Like the *VisualCafé User's Guide: Standard Features*, it contains both conceptual information and step-by-step procedures.

Portable Document Format

Portable Document Format (PDF) versions of the books are included with your copy of VisualCafé. These documents require that Adobe Acrobat

Reader be installed. Adobe Acrobat Reader is included on the installation CD-ROM of your VisualCafé product. It's also freely available from Adobe Systems at `http://www.adobe.com`.

Conventions used in this manual

This manual uses the following typographic conventions:

- Names of files, resources, classes, methods, and variables, as well as code fragments and information you type, appear in the `code typeface`. Metanames appear in *italic*. A metaname is a descriptive placeholder for a real name. For example, when referring to a project's `.vep` file, we say *projectname*`.vep`, rather than specifying a specific project name.
- Terms that appear in the glossary appear in **bold** type when they're defined in the text.
- Names of menus, menu items, buttons, and other user interface elements appear in this typeface.
- Keys you press at the same time are shown as follows: CTRL-G (press the CTRL and G keys simultaneously). Please note that even though the letter keys are listed in uppercase, you should not hold down the SHIFT key when executing these key combinations unless the SHIFT key is listed as part of the combination.
- We use the word "program" to refer to whatever you're creating with VisualCafé, whether it's an applet, application, library, or servlet.
- Wherever possible, we use the term "folder" rather than "directory" in accordance with standard Windows style, except in cases where the VisualCafé interface uses "directory." Since Windows also uses the DOS system (which primarily uses the term "directory"), and VisualCafé makes use of this DOS–Windows relationship, some areas of the product deal with "directories."

CHAPTER 2

Setting Up for the Tour

Before embarking on the tour, you need to make sure you have the necessary tools. This chapter explains how to set up the software needed for the tour.

Preparing folders for the tour projects

Your VisualCafé installation folder contains a folder named `Tour` that contains the files used in each section of the tour. You might want to create a backup of these folders or copy them to a work area, preserving the originals in the installation folder. If you need to, you can re-install the tour files from the CD at any time.

For the purposes of the tour, this document refers to your working copy of the tour folders as `Tour\`*folder name*.

Setting up for the tour

To use the tour in VisualCafé, you need to install, at a minimum, the VisualCafé program files and the tour files.

If you already installed VisualCafé with the Typical configuration, you're ready to start the tour, which begins in Chapter 3, "VisualCafé Fundamentals." Otherwise, follow the instructions on the CD case for installing the typical VisualCafé configuration.

Setting up for the tour of the database features

To use the VisualCafé database features described in Chapter 4, you must install the VisualCafé program files, the tour files, and the database software. You also need to create the tour database and connect to it. The following section explains how to set up and connect to the tour database.

When you have successfully set up VisualCafé and the database, and have connected the two, you are ready to proceed with the tour. You can start in Chapter 3 and have a look at the core features of the product, or jump directly into the database features in Chapter 4, "Accessing Your Data."

Creating the tour database

The VisualCafé tour uses a Personal Oracle Lite database called HPC. This database stores information about the orders place by the customers of the fictional Hartwig Pharmaceutical Company.

Note: For the purposes of the tour, you should install the database locally (on the same machine as VisualCafé).

To install the database server:

1. Insert the VisualCafé CD into your CD-ROM drive.

   Normally, this starts the install program.
2. If step 1 didn't start the install program, run `Setup.exe`.
3. Click Oracle Lite to launch the installer.

   Note: When installing Oracle Lite, it is advisable to use the default settings offered by the installer and let the installer update the registry for you.

   ---
4. When you are finished installing the database server, reboot the computer.

To create the database for the tour:

1. From the Windows Start menu, choose Programs, then Oracle Lite, then Oracle8 Navigator.
2. From the Oracle8 Navigator's File menu, choose New, then New Oracle Lite Database.

   The Oracle8 Navigator displays the New Oracle Lite Database Properties dialog box.
3. Enter HPC in the Name field. You can optionally enter a description.
4. Click OK to create the database and close the dialog box.
5. Right-click on the HPC icon and choose Restore from the pop-up menu.

   The Oracle8 Navigator displays an Open dialog box.
6. Navigate to the `Tour\data` folder, select `HPC.bkp` and click Open.

   The Oracle8 Navigator restores the database and displays a message box indicating that the restoration completed successfully.
7. Click OK to close the message box.
8. When you are finished, choose Exit from the File menu to close the Oracle8 Navigator.

To set up a datasource for the HPC database:

1. Launch the ODBC Data Source Administrator utility from the Windows Control Panel.
2. Click Add to display the Create New Data Source dialog box.
3. Select Oracle Lite 35 ODBC Driver and click Finish.

   This displays the Oracle Lite 3.5 ODBC 2.0 Setup dialog box.
4. Enter the following values in the fields provided:

   Data Source Name = `HPC`

   Data Description = `VisualCafe Tour Database`

   Database Directory = the full path to your Oracle Lite database directory. The database directory is called `OLDB35` and exists in your Oracle Lite installation directory.

   OLite Database = `HPC`

   Data Isolation Level = `Read Committed`

   Autocommit = <leave blank>

   Default Cursor Type = `Static`

5 Click OK to save and close the Oracle Lite 3.5 ODBC 2.0 Setup dialog box.

6 Click OK to close the ODBC Data Source Administrator utility.

After you have performed the procedures outlined above, launch VisualCafé.

Testing the database connections

After you install all the necessary tools and create the database, you're ready to get started. But before you begin the tour, you should launch the database tools and make sure all your connections are in place.

The following procedure allows you to quickly test your configuration. In most cases, you should have no trouble going through this verification process. If, however, you have any problems with this procedure, refer to the chapter on setting up data access in the *VisualCafé User's Guide: Expert Features*.

To verify your configuration:

1 Launch VisualCafé.

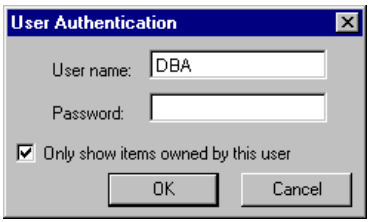
2 Choose dbNAVIGATOR from the View menu.

3 Click the plus sign (+) next to the JDBC-ODBC Bridge icon to display its data sources.

4 Now expand the `HPC` data source and its `Tables`.

If this is the first time you have connected to the database during this session, VisualCafé displays the User Authentication dialog box.

5 Enter `DBA` in the User name field and click OK (no password is necessary).



If you can see the HPC database and its tables, you are ready to turn to the next chapter and begin the tour. If, however, you were unable to complete this procedure, refer to the chapter on setting up data access in the *VisualCafé User's Guide: Expert Features*.

CHAPTER 3

VisualCafé Fundamentals

This chapter begins the VisualCafé tour. As an introduction to what you will create during the Tour, you'll first take a look at the finished product, viewing it both in VisualCafé and your Web browser. Then you'll work through the process of creating an applet and adding it to a Web page. You'll also run your applet through a debugger session, take a look at its class hierarchy, and make it localizable.

Setting the scene

Imagine that you are in charge of implementing a Java-based order-administration system for your employer, the Hartwig Pharmaceutical Company. Hartwig sells health-care products to hospitals, doctors, clinics, and pharmacies. Sales orders are generated through direct sales, manufacturers' representatives, and existing customers restocking inventory.

All sales orders are processed through the Orders department. The order-administration system is the first piece of the accounting, warehousing, and distribution system that Hartwig has decided to implement in Java. The Tour, which begins in this chapter and continues in the chapters that follow, leads you through the development of the order-administration system using VisualCafé.

Please note that Hartwig is a purely fictional company and that any similarities to any other company, real or fictional, is completely coincidental.

Taking a first look at the demo

Before jumping into the development of your Java programs, take a few moments to try out the finished version. The finished project, applet, and HTML files are in the `Tour\Complete` folder.

Opening the customer project in VisualCafé

The first step is to take a look at the `customer.vep` project in VisualCafé.

To open the customer project:

1. Launch VisualCafé.
2. Choose Open Project from the File menu.
3. Navigate to the `Tour\Complete` folder, select the `customer.vep` project and click Open.

   VisualCafé opens the project and displays the Project window. The Project window shows the applets and their components, the HTML files, and other pieces of the project.
4. Choose the Execute command from the Project menu.

   VisualCafé launches the `CustomerInfo` applet using the AppletViewer. (If this is the first time you have executed an applet since you installed VisualCafé, you will see the Java license agreement. Click Accept to close the dialog box.)
5. When you are finished test-driving the applet, choose the Quit command from the Applet menu.

Opening the CustomerInfo Web page in a Web browser

Finally, you can open the Web page as the user will see it — in a Web browser.

To see the Web page displayed in a Web browser:

1. From the Project menu, choose Options.

   VisualCafé displays the Project Options dialog box.
2. Click the Project tab.
3. Check the Execute applet in default Web browser checkbox.

4. Select the `CustomerInfo.html` file in the Start with Web page drop-down list.
5. Click OK to set the options and close the dialog box.
6. Choose Execute from the Project menu.

   VisualCafé launches your Web browser to display the Web page.
7. When you are finished viewing the applet, close your browser.

Now that you've seen the applet in action, close the project.

To close the project:

- Choose Close Project from the File menu.

In the next section, you'll start building the project yourself.

Creating the Customer Information applet

The Customer Information (`CustomerInfo`) applet is one piece of the customer project. As you have seen, this project also contains a dialog box and an HTML file. However, the first step in VisualCafé development is to create and save the project.

In this section, you will:

- Create and save a project
- Create a JFC (Swing) applet
- Add Swing components to the applet
- Create interactions to make the components respond to user actions

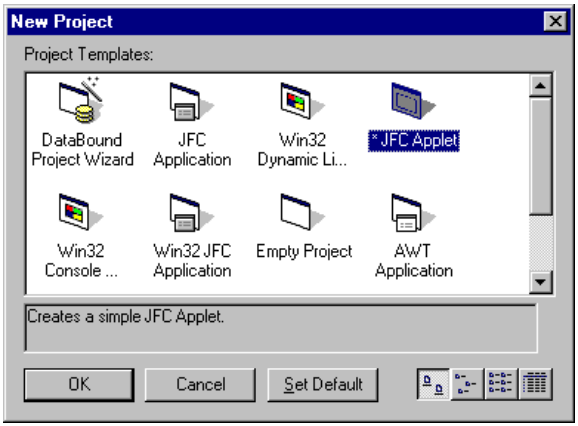
Creating the customer project

In VisualCafé, you use projects to organize the files you need for your application and applet development.

To create the customer project:

1 Choose New Project from the File menu.

VisualCafé displays the New Project dialog box.



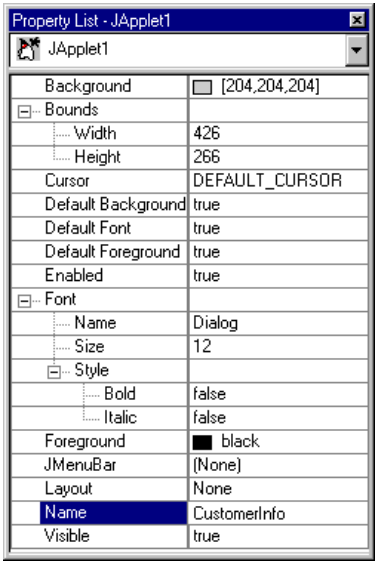
The project templates in the New Project dialog box automate many of the steps of creating different kinds of projects. See the *VisualCafé User's Guide: Standard Features* for information about project templates.

2 Select JFC Applet and click OK.

VisualCafé creates a project that contains a JFC applet, displays the Project window, and opens the applet in the Form Designer.

3 Click on the `JApplet1` icon in the Project window.

When you select an object in the Project window, its properties appear in the Property List:



4 In the Property List, click on the Name field and replace the default name `JApplet1` with the name `CustomerInfo`.

The Property List is a powerful tool for modifying the characteristics of objects. The left column lists the property names, while the right column gives the values. Some of the right-column fields are text fields, some are drop-down lists, and some invoke more complex customizers like color pickers.

You can display the Property List for an object in many ways. For example, you can pull down the list at the top of the Property List window and choose from a list of the objects in the project, you can right-click on an object and select Properties from the pop-up menu, or you can select and object and then just look in the Property List. See the *VisualCafé User's Guide: Standard Features* for more information about the Property List.

Saving the project

It's always a good idea to save your projects often. So, before you go any further, save the project.

To save the customer project:

1. Click on the Project window.

   You need to tell VisualCafé what you want to save. Clicking on the Project window says that you want to save the project.
2. Choose the Save As command from the File menu to display the Save As dialog box.
3. Navigate to the `Tour\MyProject` folder.
4. Enter `customer` in the File name field and click Save.

   VisualCafé automatically appends the `.vep` extension for you.

Adding components to the applet

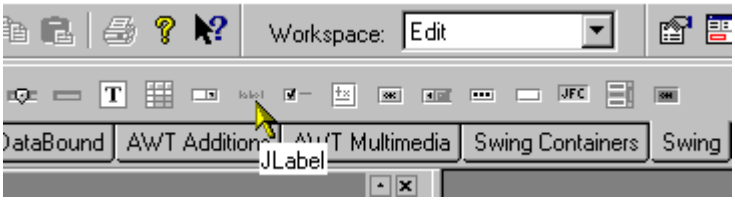
Now that you have created a project with an applet, you can start adding the necessary components. Begin by adding a `JLabel` to the applet.

To add a JLabel to the applet:

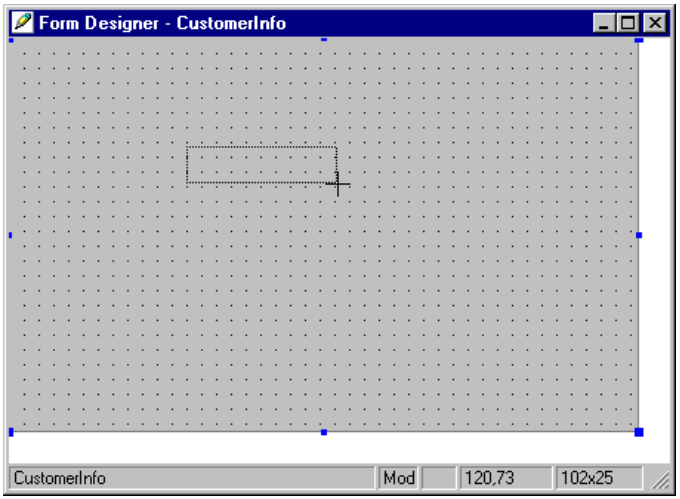
1. Click the Swing tab on the Component Palette.

The Component Palette gives you easy access to components, so that you can click on them and drop them into your project. It shows many, though not all, of the components that are in the component library. You can add components to the palette, or remove components from the palette, by using the Environment Options command from the Tools menu. See the *VisualCafé User's Guide: Standard Features* for more information on the Component Library and Component Palette.

2 Click the `JLabel` component.



3 Click on the applet in the Form Designer at the place you want to put the `JLabel` and drag to set its size and shape.



You use the Form Designer to build your user interface by dropping components into your forms. In this case, the form is the applet.

The Form Designer has various tools and settings that help you create the user interface that you want. It also gives you visual access to your components — you can click on a component to display the component's properties, and when a component is selected you can use menu commands to cut or copy it. See the *VisualCafé User's Guide: Standard Features* for more information on the Form Designer.

4 In the Property List, enter `Cust ID:` as the value of the Text property.

To add a JTextField to the applet:

1. Click the `JTextField` component (on the Swing tab of the Component Palette), drag it onto the Form Designer, and drop it on the applet.
2. In the Property List, enter `cust_num` in the Name field.

To add a JButton to the applet:

1. Click the `JButton` component, drag it to the Form Designer, and drop it on the applet.
2. Position the `JButton` near the bottom of the applet.
3. In the Property List, set the following properties for the `JButton`:

Width:	`160`
Height:	`24`
Name:	`SelectButton`
Text:	`Select Customer ...`

Resizing and moving components

Before creating the rest of the fields and labels, resize the two components you've already created to a common height and arrange them neatly on the form.

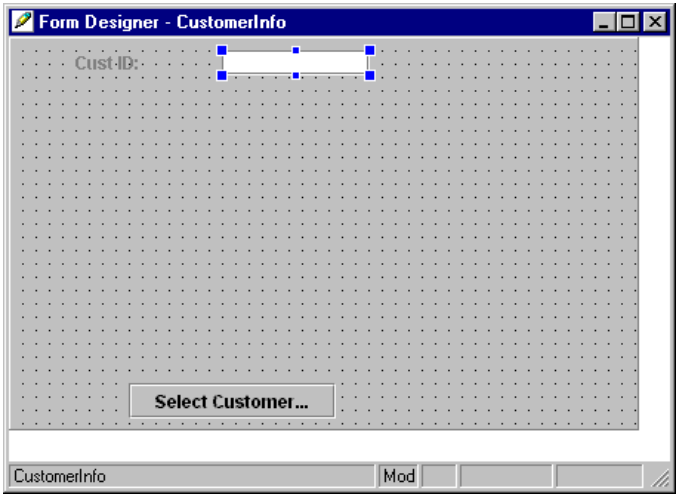
To resize and arrange the components:

1. Select both the `JLabel` and the `JTextField` by clicking one and then Shift-clicking the other.
2. In the Property List, enter `99` in the Width field and `17` in the Height field.

   This sets the Width field and Height values for both components. As you can see, when you select multiple components, the Property List shows only the properties they share.
3. Drag the `JLabel` to a position near the top of the form.
4. Drag the `JTextField` to align it horizontally with the `JLabel`.

   You can move components a pixel at a time by selecting them and using the arrow keys. If they move more than a pixel, you may have

the “snap to grid” option set. You can change this setting using the Grid Options command of the Layout menu.

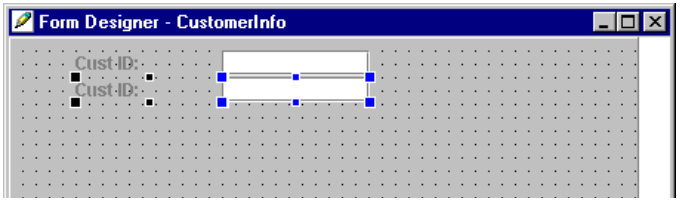


Copying components

Using this set of two components, create nine more sets (for a total of ten `JLable`s and 10 `JTextField`s).

To create duplicates of these components:

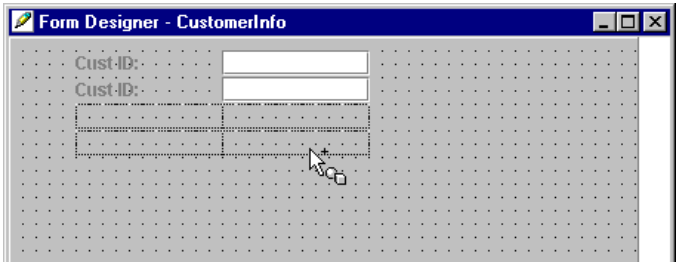
1. Grab both the `JLabel` and the `JTextField` by clicking one and then Shift-clicking the other.
2. Choose Copy, then Paste from the Edit menu.
3. Drag the new set to align it vertically under the first set.



That is one way to duplicate components, here’s another.

To select and duplicate all four components:

1. Click and drag out a selection box around all four.
2. Control-drag (hold down the Control key and drag with the mouse) the components to the position you want to paste the new ones.



3. Now create six more sets.

Now that you've added all the components, arrange them on the form the way you like. You can use the arrow keys and the commands in the layout menu to help you make the arrangement neat.

To adjust the components:

1. Change the Name and Width properties of the `JTextField`s and the Text property of the `JLabel`s to the values shown in the following table.

JLabel Text	JTextField Name	JTextField Width
`Cust ID:`	`cust_num`	`60`
`Customer:`	`cust_name`	`180`
`Address:`	`address`	`180`
`City:`	`city`	`120`
`State:`	`state`	`60`
`Postal code:`	`postal_code`	`120`
`Country:`	`country`	`120`
`Contact:`	`contact`	`120`
`Phone:`	`phone`	`120`
`Status:`	`status`	`60`

2. When you are finished editing properties, click the Project window and choose Save All from the File menu, to save the project file and the applet's .`java` file.

Adding a JMaskedTextField

The text that will appear in the Phone field is simply a 10-digit number. To make this text more readable replace the `JTextField` named `phone` with a `JMaskedTextField`.

To add the JMaskedTextField:

1. Delete the `phone JTextField`.
2. Click the JMaskedTextField component (on the Swing Additions tab of the Component Palette), drag it to the Form Designer, and drop it on the applet.
3. In the Property List, enter `phone` in the Name field.
4. Re-position it to where the `JTextField` was.
5. In the Property List, enter `999-999-9999` in the Mask field.
6. Save your changes.

Running the Customer Information applet

Now that you have the basic user interface put together, take a moment to run the applet and see how it works.

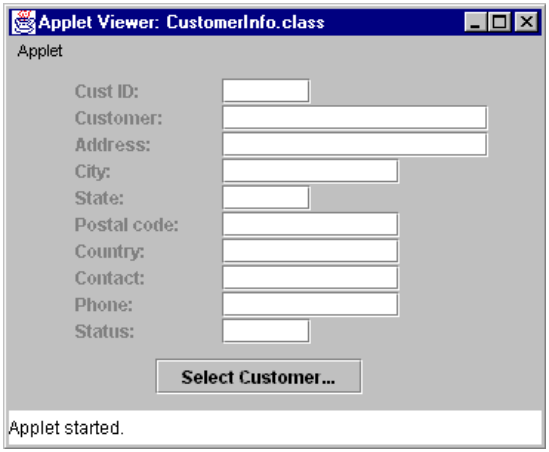
Running the applet

You can test your applets by running them within the VisualCafé environment.

To run the Customer Information applet in VisualCafé:

1. Click the Project window.
2. Choose Execute from the Project menu.

   VisualCafé runs the applet in the AppletViewer.



3. Close the applet by choosing Quit from its Applet menu.

Adding the CustomerList dialog box

At this point, your project includes an applet and a set of components that make up its user interface. Now you need to add data to the applet, create interactions that will cause that data to display, and include a Web page in the project, in which the finished applet will be executed.

Note: If you would like to start here without completing the previous section, you can start this section with the `customer.vep` project in `Tour\T1`.

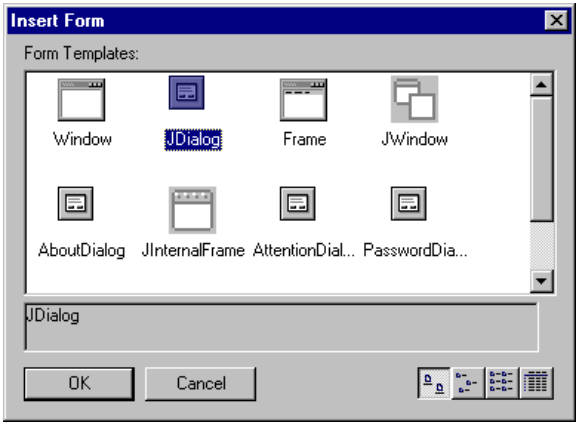
Adding the dialog box

The dialog box you are about to create allows the user to select the customer whose information they want to display.

To add a dialog box to the project:

1. Click on the `CustomerInfo` applet in the Project window.
2. Choose Form from the Insert menu.

   VisualCafé displays the Insert Form dialog box.



3. Select `JDialog` and click OK.

Adding components to the dialog box

The finished dialog box will contain several components, including:

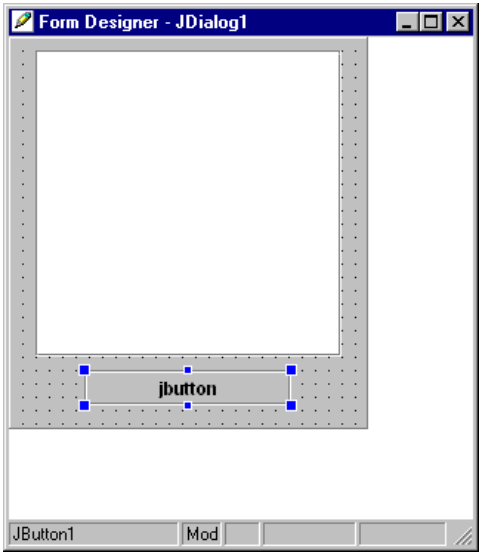
- A `JScrollPane` component
- A `Jlist` component
- A `JButton` component

To add the components to the dialog box:

1. Click the `JScrollPane` component (on the Swing Containers tab of the Component Palette), then click-and-drag in the Form Designer to draw out its shape.
2. Click the `JList` component (on the Swing tab), drag it to the Form Designer, and drop it in the `JScrollPane`.

   You don't need to resize the `JList`, it will automatically resize to fit the `JScrollPane`.
3. Finally, drag out a `JButton` on the form.

   You should have something like this:



4. Now make the following changes to the properties of the `JDialog` components:

	`JDialog`	`JScrollPane`	`JList`	`JButton`
Name	`CustomerList`	`ListPane`	`PickList`	`OkButton`
Width	`243`	`207`		`140`
Height	`265`	`207`		`24`
Text				`OK`
Title	`Select a Customer`			

Adding interactions

At this point you need to add an interaction for each of the buttons, so that when the user clicks the Select Customer button the applet displays the dialog box, and when the user clicks OK the dialog hides itself.

When you add an interaction, an event such as a mouse click triggers something, typically a change to another component. In this case, a mouse click on one of the buttons either shows or hides the `JDialog` component. See the *VisualCafé User's Guide: Standard Features* for more information on working with events and interactions.

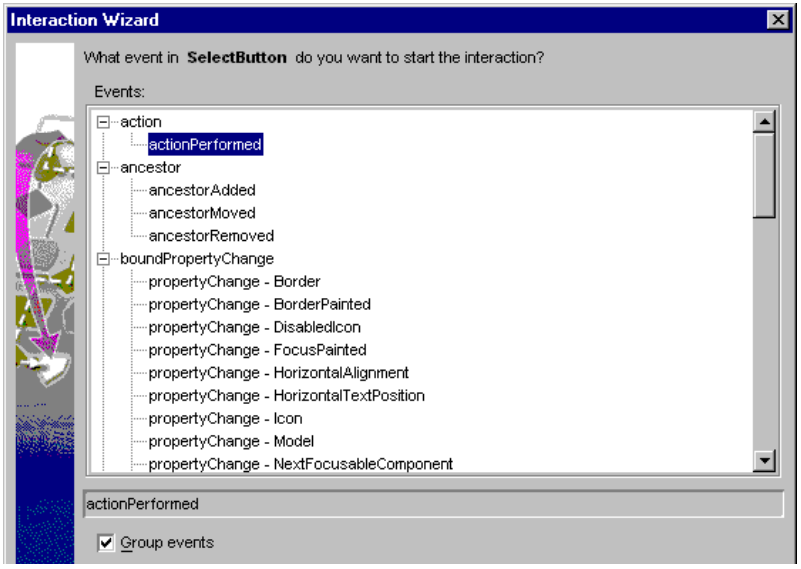
To add the interaction for the SelectButton:

1. Click on the `CustomerInfo` applet in the Project window.
2. Right-click on the `SelectButton` in the Form Designer.
3. Choose Add Interaction from the pop-up menu.

   You can also find this command in the Object menu.

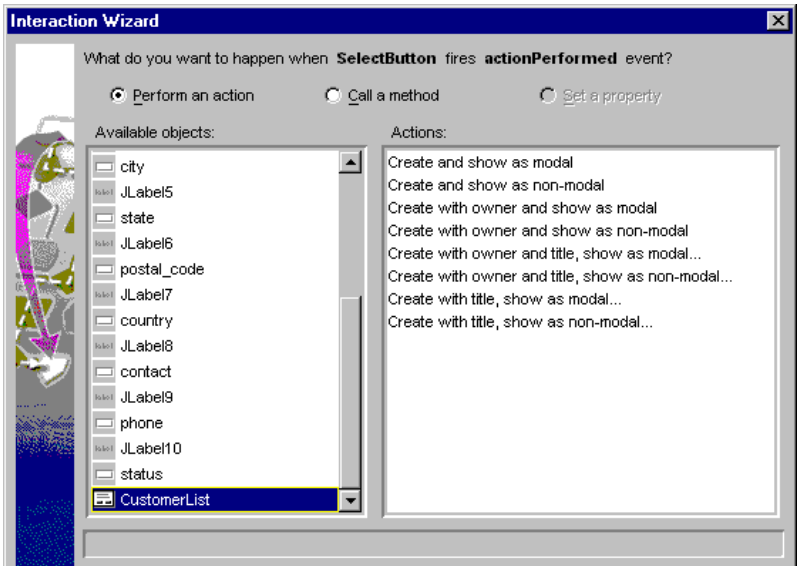
   VisualCafé launches the Interaction Wizard. The first page gives you some information about the wizard.
4. Click Next.

   The wizard prompts you to select an event to start the interaction.



5. Choose actionPerformed, if it is not already chosen, and click Next.

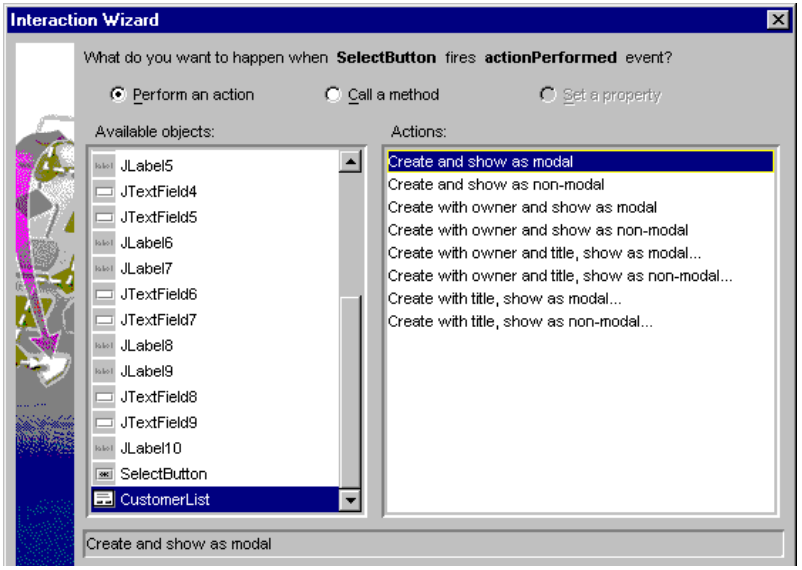
The wizard prompts you to select what you want to have happen.



6 Click the Perform an action radio button if it is not already chosen.

7 Select CustomerList from the list of Available objects.

8 Select Create and show as modal from the list of Actions.



9 Click Next.

VisualCafé displays a summary of your selections.

10 Click Finish.

VisualCafé generates the code for the interaction based on your selections. You will now see an interaction arrow on the Form Designer window.

11 Click on the Project window and choose Save All from the File menu to save your work.

To add the interaction for the OkButton:

1 Click on the `CustomerList` dialog in the Project window.

2 Right-click on the `OkButton` in the Form Designer.

3 Choose Add Interaction from the pop-up menu.

4 Choose actionPerformed and click Next.

5 Click the Perform an action radio button.

6 Select CustomerList from the list of Available objects.

7 Select Hide the CustomerList from the list of Actions.

8 Click Finish.

VisualCafé generates the code for the interaction based on your selections. You will now see an interaction arrow on the Form Designer window.

9 Click on the Project window and choose Save All from the File menu to save your work.

Adding some sample data

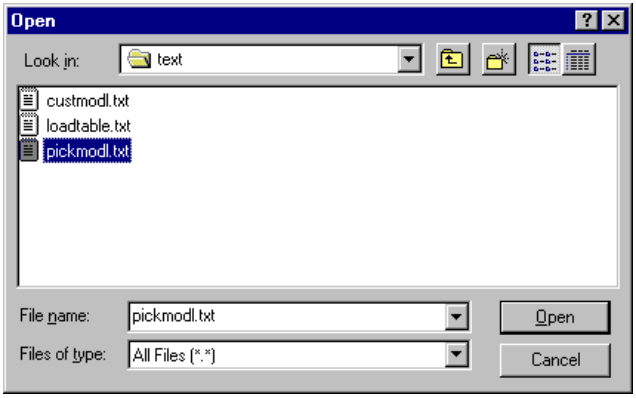
You have created the visual components to display the data. Swing components, which are the type used in this tour, use models to store data. (See the Swing chapter of the *VisualCafé User's Guide: Standard Features* for more information on Swing models.) You now need to create models to hold the data. Then you will fill the models with data. In the `Tour\data\text` folder, you'll find the following three files:

- `custmodl.txt` contains data for the `CustomerInfo` applet.
- `pickmodl.txt` contains data for the `CustomerList` dialog.
- `loadtable.txt` contains a piece of code you will use in an upcoming section.

You will need a model for both the applet and the dialog box. In the `CustomerList` dialog, the `PickList` component will display its data from a `StringListModel`. In the `CustomerInfo` applet, the fields will display data from a `StringTableModel`.

To add a StringListModel to the CustomerList dialog:

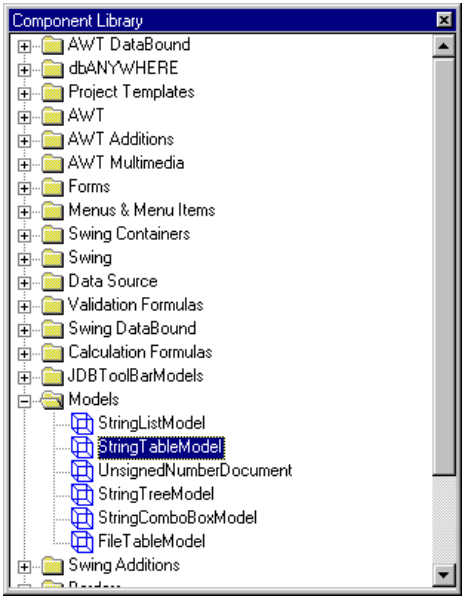
1. Click on the `PickList` component in the Project window.
2. In the Property List, choose new StringListModel from the Model property's drop-down list.
3. In the Project window, click on the model VisualCafé created and change its Name property to `PickModel`.
4. Choose Open from the File menu and navigate to the `Tour\data\text` folder.
5. Choose All Files (*.*) from the Files of type drop-down list.



6. Select `pickmodl.txt` and click Open.
7. Choose Select All from the Edit menu to select the entire contents of the file, then Copy from the Edit menu to copy it.
8. Click on the `PickModel` component in the Project window.
9. Click on the Items field in the Property List.
10. Press Control-V to paste the data into the Items list.
11. Close the Source window that contains the text file.
12. Click on the Project window and choose Save All from the File menu to save your work.

To add a StringTableModel to the CustomerInfo applet:

1. Choose Component Library from the View menu to display the Component Library window.
2. In the Component Library window, find the Models folder.



3. Drag the `StringTableModel` icon from the Models folder and drop it on the `CustomerInfo` applet.
4. In the Project window, click on the `StringTableModel` and change its Name property to `CustModel`.
5. Open the `custmodl.txt` file in the `Tour\data\text` folder.
6. Select the entire contents of the file, then copy it and paste the data into the `CustModel` Items list.
7. Close the Component Library window and the Source window that contains the text file.
8. Click on the Project window and choose Save All from the File menu to save your work.

Adding code to the applet

In the `Tour\data\text` folder, you'll find the `loadtable.txt` file, which contains a small section of Java code. This piece of code defines a

method that you will use to populate the fields of the applet. You need to add this method to the applet, and then call the method in the interaction for the `SelectButton`.

To add the method to the applet:

1. Open the `loadtable.txt` file, select the entire contents of the file, and copy it.
2. Right-click on the applet and choose Edit Source from the pop-up menu to open a Source window.
3. Scroll down to the end of the file.
4. Paste the contents of the `loadtable.txt` file immediately before the last closing brace in the file.
5. Press Control-S to save changes.
6. Close the Source window that displays the `loadtable.txt` file.

To add the method call to the interaction:

1. In the Source window that displays the code for the `CustomerInfo` applet, choose SelectButton from the Objects drop-down list.
2. Choose actionPerformed from the Events drop-down list.
3. Scroll down until you see the following section of code:

```
try {
    // CustomerList Create and show as modal
    {
        CustomerList CustomerList1 = new CustomerList();
        CustomerList1.setModal(true);
        CustomerList1.show();
    }
} catch (java.lang.Exception e) {
}
```

4. Enter the code to call the method as follows:

```
try {
    // CustomerList Create and show as modal
    {
        CustomerList CustomerList1 = new CustomerList();
        CustomerList1.setModal(true);
        CustomerList1.show();
        loadTable(CustomerList1.PickList, this.CustModel);
        this.requestFocus();
    }
} catch (java.lang.Exception e) {
}
```

5. Press Control-S to save changes.
6. Close the Source window.

If you'd like to see how the applet works so far, choose Execute from the Project menu to run it. When you are finished, close the applet by choosing Quit from its Applet menu.

Adding an HTML file to the project

So far, when you execute the project, VisualCafé generates a skeleton HTML file and uses the AppletViewer to display it. Now you need to add a more finished-looking HTML file to display your applet. Then you can specify VisualCafé to open the custom HTML file in the default Web browser.

In the `Tour\files` folder, you'll find the `CustomerInfo.html` file. Copy this file to your project folder before adding it to the project.

To add an HTML file to the project:

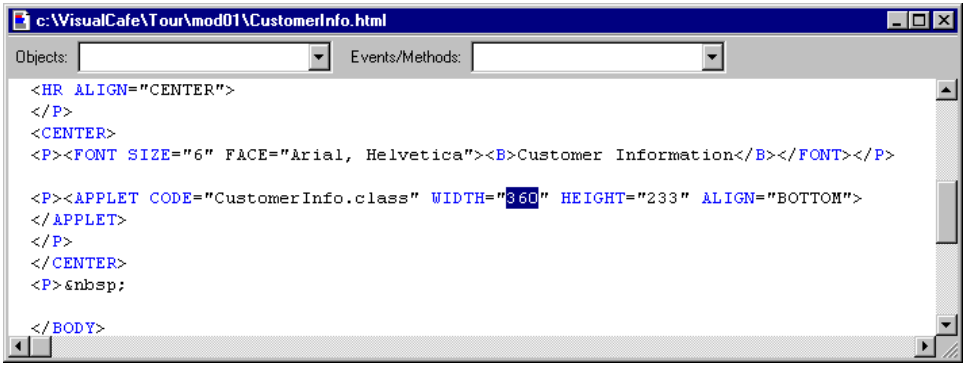
1. Click the Files tab on the Project window.
2. Right-click in the Project window and choose Insert Files from the pop-up menu.
3. Navigate to your project folder.
4. Select Net files from the Files of Type drop-down list.
5. Select the `CustomerInfo.html` file and click Add.
6. Click OK.
7. Click on the Project window and choose Save All from the File menu to save your work.

Edit the applet tag to fit your applet:

1. Double-click on the HTML file in the Files tab on the Project window.

   VisualCafé opens a Source window to display the file.

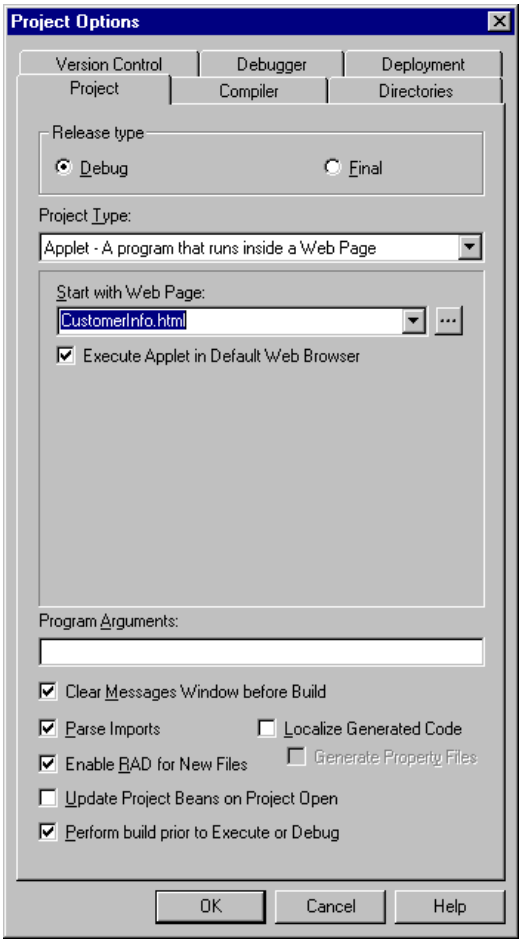
2. Find the `WIDTH` and `HEIGHT` parameters of the applet in the HTML file and change them to match the actual width and height of the applet.



3. Press Control-S to save changes.
4. Close the Source window.

Edit the project options to launch the custom HTML page in the default Web browser:

1. Choose Options from the Project menu to open the Project Options dialog box.
2. On the Project tab of the Project Options dialog box, check the Execute applet in default Web browser checkbox.



3. In the Start with Web Page list box, select `CustomerInfo.html`.
4. Click OK to apply changes and close the dialog box.
5. Click on the Project window and choose Save All from the File menu to save your work.

6 Choose Execute from the Project menu to run the applet and see how it runs in your browser with the HTML file. When you are finished choose Close from the File menu to close the browser.

Note: To run Java applets in your browser, the browser must support JDK 1.1. See the VisualCafé `ReadMe.html` file for information on running Java applets in Netscape Navigator and Microsoft Internet Explorer.

Deploying the customer project

With VisualCafé, the process of deploying a project is simply specifying where and how to deploy, modifying the HTML file, and choosing a menu item.

Note: If you would like to start here without completing the previous section, you can start this section with the `customer.vep` project in `Tour\T2`.

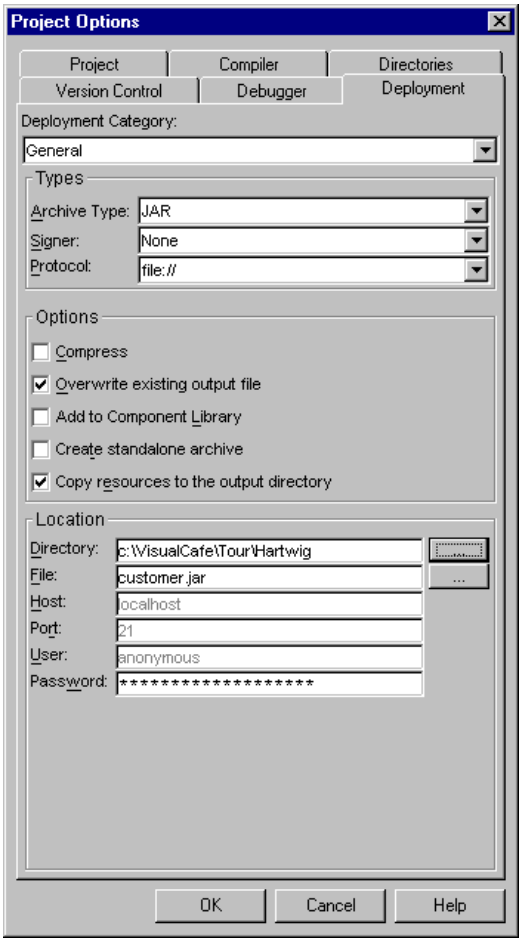
To specify the deployment options:

1 Choose Options from the Project menu to display the Project Options dialog box.

2 Click the Deployment tab to display deployment options.

3 Make sure General is selected from the Deployment Category drop-down list.

4 Make sure Archive Type is set to `JAR` and Protocol is set to `file://`.



5 In the Location group, enter the full path to the `Tour\Hartwig` folder in the Directory field and specify `customer.jar` as the File.

6 Click OK to apply your changes and close the dialog box.

To modify the HTML file:

1 Right-click on `CustomerInfo.html` in the Files tab of the Project window.

2 Choose Edit Source from the pop-up menu.

3 Edit the applet tag to include an `ARCHIVE` variable for the `customer.jar` file.

```
<APPLET ARCHIVE="customer.jar" CODE="CustomerInfo.class" ...
```

4. Press Control-S to save changes.
5. Close the Source window.

To deploy the project:

- Choose Deploy from the Project menu.
  VisualCafé generates the classes for the project, packages them as you have specified, and writes them to the location you have specified.

To run the deployed project:

1. Launch your browser and open the `CustomerInfo.html` file in the `Tour\Hartwig` folder.
2. When you are finished choose the Exit command from the File menu to close the browser.

Debugging the CustomerList dialog

Although the `CustomerList` dialog box works properly in your application, you can explore some of VisualCafé's debugging features by working with its source code.

Note: If you would like to start here without completing the previous section, you can start this section with the `customer.vep` project in `Tour\T3`.

To start debugging the CustomerList dialog box:

1. If the `customer.vep` project isn't already open, open it now.
2. Choose Options from the Project menu to display the Project Options dialog box.
3. On the Project tab of the Project Options dialog box, deselect the Execute applet in default Web browser checkbox.
4. Select (Automatic) in the Start with Web page drop-down list.
5. Make sure Release type is set to Debug.
6. Click OK to apply changes and close the dialog box.

7. Click on the Project window and choose Save All from the File menu to save your work.
8. Choose Run in Debugger from the Project menu.

   VisualCafé launches the applet.

   Note: When your application is running in the debugger, `[Executing]` appears in VisualCafé's title bar. When the applet is paused it does not redraw its window, so its appearance can be somewhat erratic until execution continues.

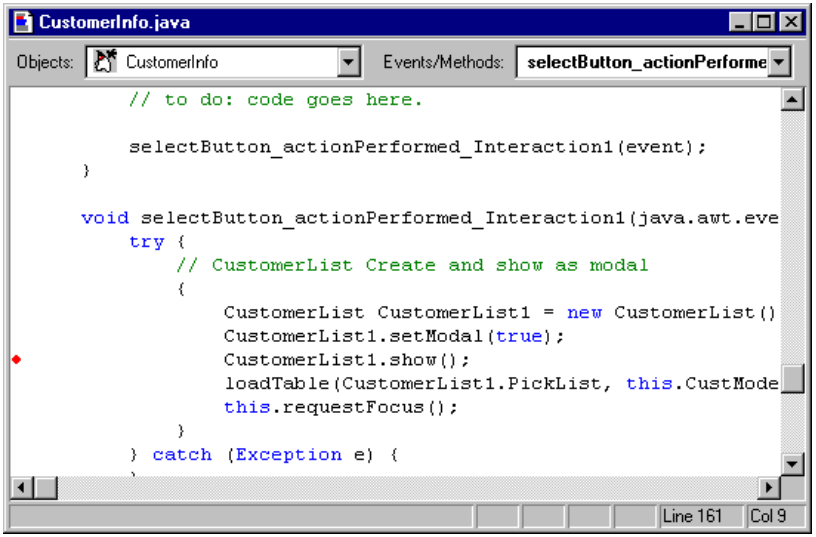
   The Variables, Call Stack, and Breakpoints windows should be displayed. If not, you can open each window by choosing the corresponding menu item from the View menu.
9. Choose Watch from the View menu to display the Watch window.

   You can use the Watch window to display the value of selected variables.

To set a breakpoint:

1. In the Project window, right-click on `CustomerInfo` and choose Edit Source from the pop-up menu.
2. Choose SelectButton from the Objects drop-down list and actionPerformed from the Methods drop-down list.
3. Scroll down until you see the `CustomerList1.show()` method.

4 Right-click the line with the show method and choose Set Breakpoint from the pop-up menu.



The breakpoint appears in the Breakpoints window and a red diamond in the Source window indicates that the breakpoint is set.

Now that you have set the breakpoint, you can trigger it and step through the code.

Note: While you are debugging, VisualCafé sometimes covers the running applet. You might need to temporarily minimize VisualCafé to interact with the applet.

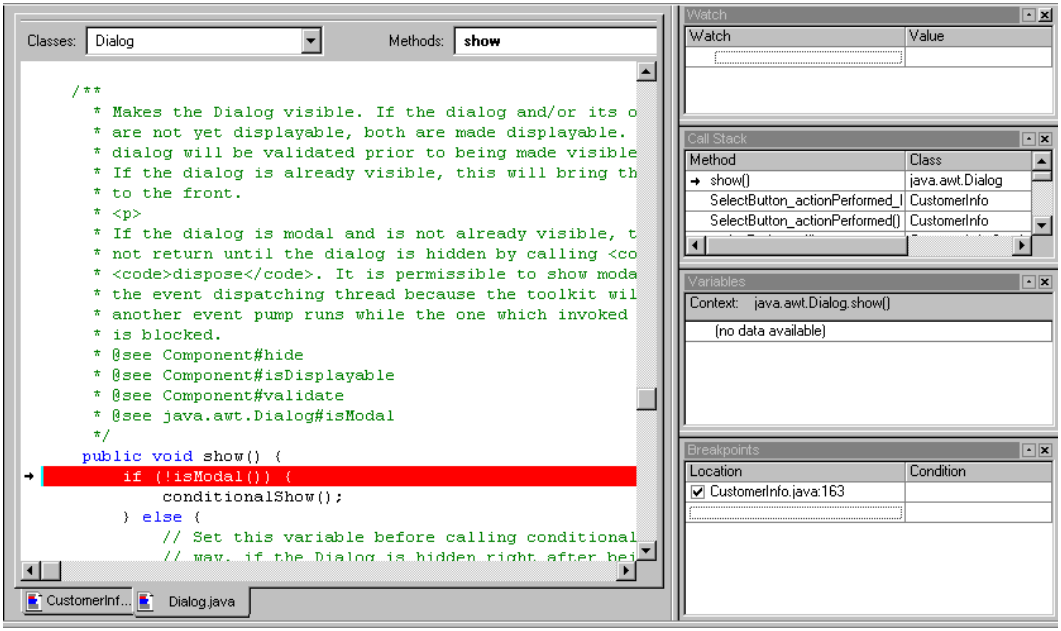
To trigger the breakpoint and step through the code:

1 In the running applet, click the Select Customer button.

The breakpoint is triggered and the breakpoint line in the Source window is highlighted. Variables and their values at this point in the code appear in the Variables window; the current call stack appears in the Call Stack window. The title bar displays `[Paused]`.

2 From the Debug menu, choose Step Into.

VisualCafé launches another Source window to display the `Dialog` code.



3. Choose Step Over from the Debug menu.

   The next call is highlighted.

4. Choose Step Out from the Debug menu.

   VisualCafé returns control to the running applet.

5. Select a customer and click OK.

   VisualCafé returns focus to the `CustomerInfo` file.

6. Right-click the breakpoint line and choose Clear Breakpoint to remove the breakpoint.

7. Choose Continue from the Debug menu.

   Now the applet resumes execution and the customer information is displayed.

8. When you are finished, choose Quit from the Applet menu to close the applet.

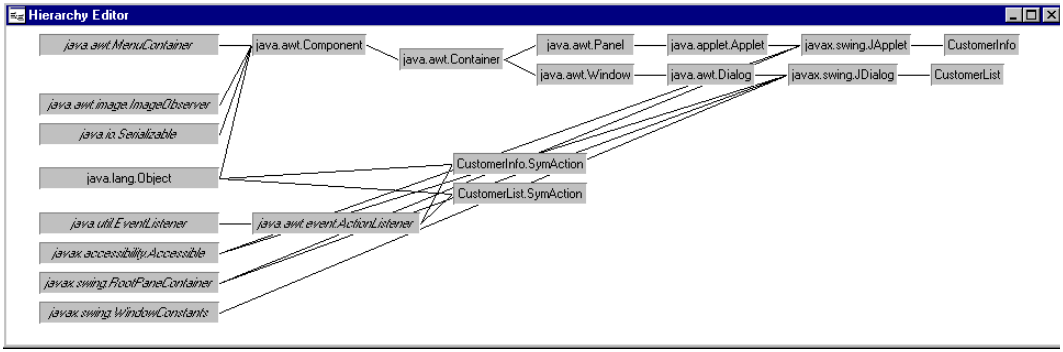
Viewing the classes used in the applet

VisualCafé has object programming tools that make it easy to understand and manage Java classes. Now that you've completed the first phase of the tour, take a moment to look at your project through the Hierarchy Editor and the Class Browser.

To view your project with the Hierarchy Editor:

1. Click on the Project window.
2. Choose Hierarchy Editor from the View menu.

   VisualCafé displays the class hierarchy of your project in the Hierarchy Editor.



   Using the Hierarchy Editor, you can modify and extend the class hierarchy for your Java programs graphically with simple drag-and-drop techniques.

3. Now double-click on the CustomerInfo class in the upper right corner of the Hierarchy Editor.

   VisualCafé displays the Class Browser.

   Using the Class Browser, you can view and edit the classes and members in your Java programs.

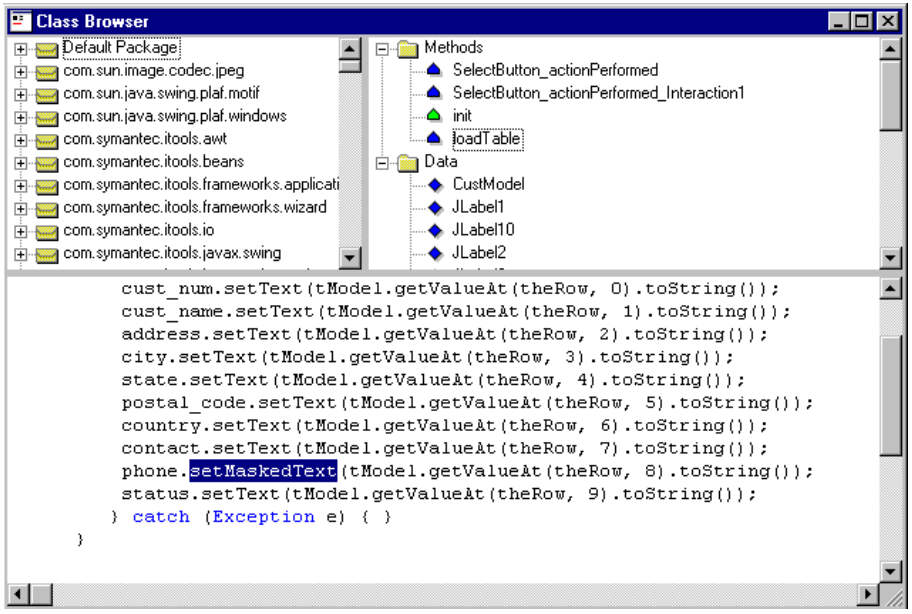
   You may have noticed that the `JMaskedTextField` called `phone`, which doesn't seem to be using the mask you gave it. That is because the proper command to set the text for a `JMaskedTextField` is `setMaskedText()`. You can use the Class Browser to fix this.

4. Click on the `loadTable` method in the Members pane (upper right).

The lower pane of the Class Browser displays the code for just the selected member. This makes it easy to edit code in member methods without accidentally making changes outside the scope of the selected object. The color-coded syntax also makes it easy to read the code.

5 Scroll down until you see the `phone.setText` line.

6 Change the statement to read `phone.setMaskedText`.



7 When you are finished, close the Class Browser and the Hierarchy Editor.

Preparing the project for localization

Since Hartwig is an international company, you need to prepare the project to be localized for the different locales that it will be used in. Localization is the process of modifying the program to fit another language or culture. In its simplest form, this means translating the text that appears in the user interface.

VisualCafé simplifies the process for you by generating resource bundles for generated code. The resource bundle stores the localizable resources

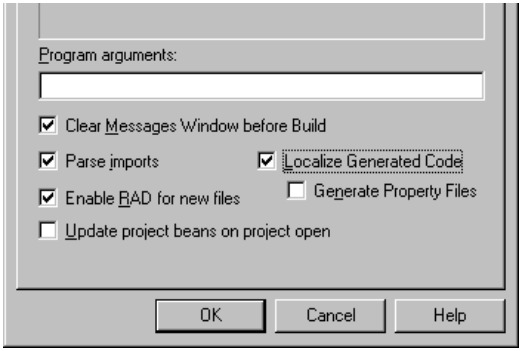
(primarily text strings) separately from the Java code. VisualCafé also provides an editor for the resource bundle to make the translation process easier.

To generate localizable resource bundles:

1. Click on the Project window and choose Options from the Project menu.

   VisualCafé displays the Project Options dialog box.

2. On the Project tab, check the Localize Generated Code checkbox.



   VisualCafé displays a message box to let you know that this will bundle resources only for the code that VisualCafé generates, and not code that you have written.

3. Click OK to close the message box.

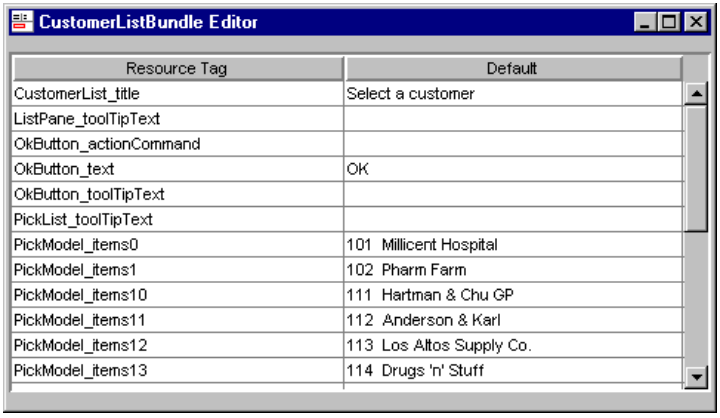
4. Click OK on the Project Options dialog box to apply changes and close the dialog box.

   VisualCafé regenerates the code for the project and creates a resource bundle for each `.java` file. VisualCafé will continue to maintain these files as long as the option is specified in the Project Options dialog box.

Now take a look at one of the resource bundles using the editor:

1. Click on CustomerList in Project window.
2. From the Tools menu, choose Localization, then Edit Resource Bundle.

   VisualCafé displays the resource names and values for the `CustomerList` dialog in the Resource Bundle Editor. You can use this editor to edit the values in the resource bundles.



3. When you are finished, close the editor.

What's Next?

You've made it through the first part of the VisualCafé tour. The tasks you have completed along the way were a small sample, intended to whet your appetite for further exploration of VisualCafé. For a tour of the database features of VisualCafé, continue to the next chapter.

CHAPTER 4

Accessing Your Data

This chapter demonstrates the features of VisualCafé that allow you create databound forms for your Java applets and applications. Here you will create an Order-Administration applet that connects to a database.

Important: Before starting this chapter, make sure your database tools and connections are set up as described in Chapter 2, "Setting Up for the Tour."

Creating the CurrentOrders applet

`CurrentOrders` (the Order-Administration applet) is the first piece of the `orders` project. This project will eventually contain other applets and applications for the accounting, warehousing, and distribution system.

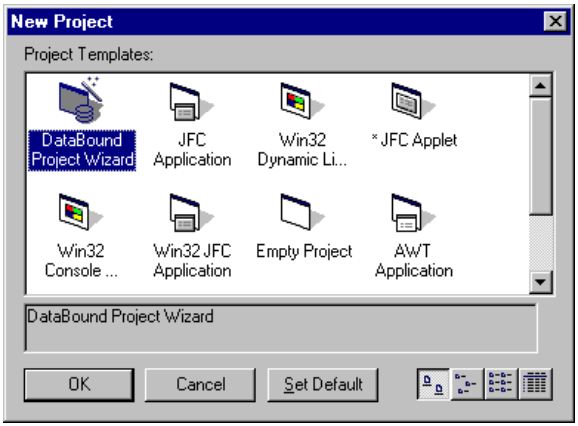
Creating the orders project

The first step to implementing the system is to create the project, then you can add the other pieces.

To create the orders project:

1. Choose New Project from the File menu.

   VisualCafé displays the New Project dialog box.



2. Select DataBound Project Wizard and click OK.

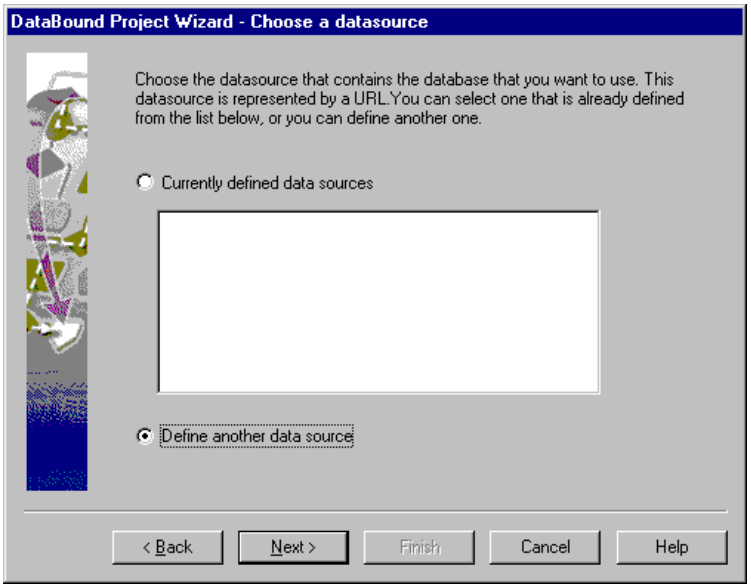
   VisualCafé launches the DataBound Project Wizard, which will guide you through the creation of the project.

3. Click Next on the Introduction page to continue.

4. On the Project Type page, select JFC Applet and click Next.

   The wizard displays the Choose a datasource page.

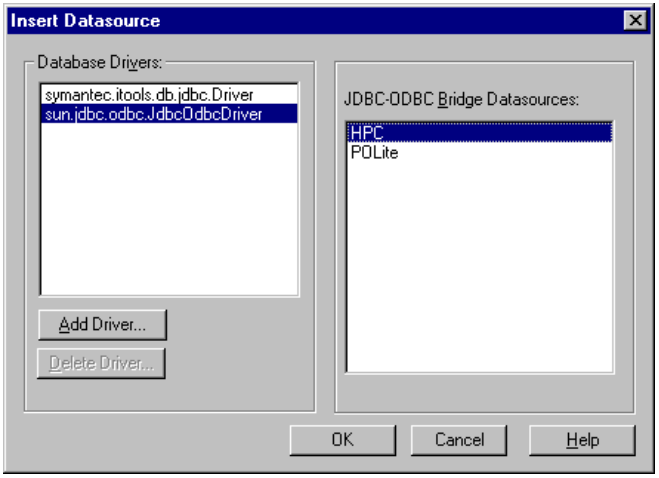
5 Click the Define another data source radio button, then click Next.



6 On the Define a Datasource page, click New.

7 In the Insert Datasource dialog box, select `sun.jdbc.odbc.JdbcOdbcDriver`, and enter the following:

JDBC-ODBC Bridge Datasources = HPC

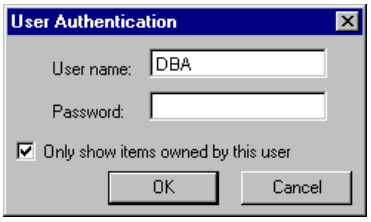


8 Click OK to select the data source and close the dialog.

9 Click Next on the Define a Datasource page.

If this is the first time you have connected to the database during this session, VisualCafé displays the User Authentication dialog box.

10 Enter `DBA` in the User name field and click OK (no password is necessary).



11 On the Choose a Table page, select the `ORDERS` table and click Next.

12 On the Choose Columns page, make sure all the columns are in the Used columns area and click Next.

The wizard displays the Choose Components page.

13 Choose `Individual components` from the Component arrangement drop-down list and modify the labels and components as follows:

Field	Label	Component
`ORDER_NUM`	`Order:`	`JTextField`
`CUST_NUM`	`Cust ID:`	`JTextField`
`ORDER_DATE`	`Date:`	`JMaskedTextField`

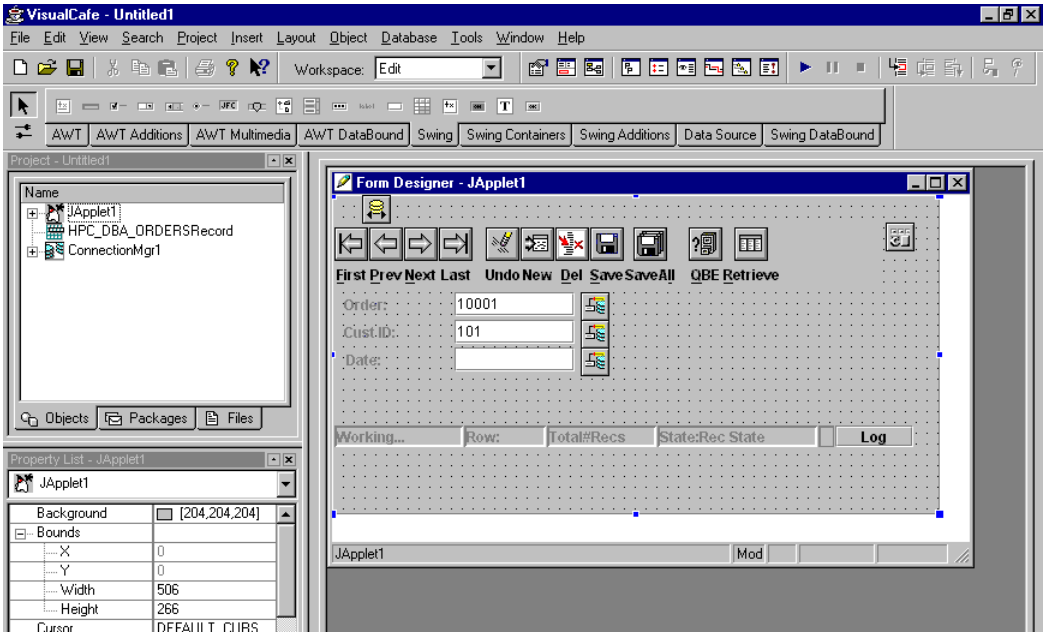
14 Click Next.

15 On the Layout Options page, leave Toolbar and Statusbar checked and click Next.

The wizard displays the Summary page that shows what you have selected.

16 Click Finish to create the project and its initial form.

VisualCafé connects to the database and generates the project.



To rename the applet and save the project:

1 Click on the `JApplet1` icon in the Project window.

2 In the Property List, click on the Name field and enter the name `CurrentOrders`.

3 Click on the Project window and choose Save As from the File menu.

4 Navigate to the `Tour\MyDBProject` folder.

5 Enter `orders` in the File name field and click Save.

VisualCafé automatically appends the extension `.vep` when you save the project.

Adding the detail tables

In the HPC database the `ORDERS` and `ITEMS` tables work together in a master-detail relationship. You already have the `ORDERS` table represented

on the applet; now you need to add its detail table, `ITEMS`. Then you will add the `CUSTOMER` table to display customer information on the form.

Adding a detail table with the Add Table Wizard

To add the ITEMS table:

1. Choose Add database table from the Database menu.

   VisualCafé launches the Add Table Wizard.

2. Click Next on the Introduction page to continue.

   The wizard displays the Choose a Datasource page.

3. Select the HPC data source and click Next.

4. On the Choose a Table page, select the ITEMS table and click Next.

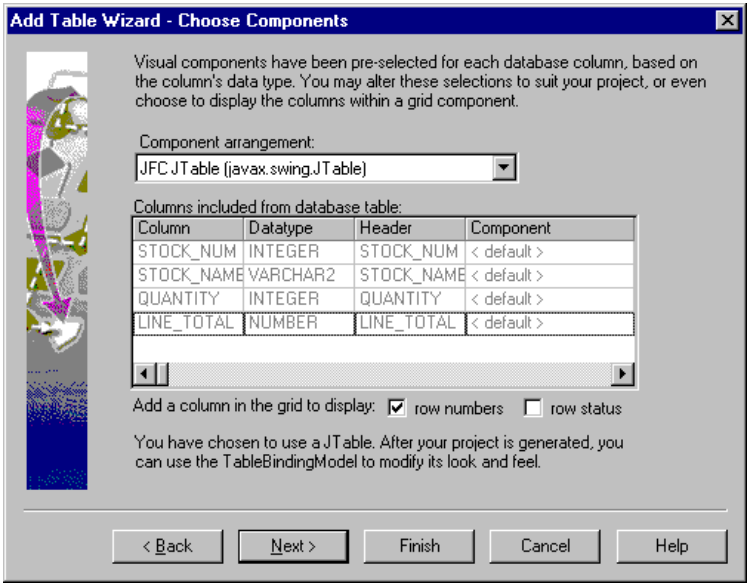
5. On the Choose Columns page, select ORDER_NUM in the Used columns list and click Remove.

6. Remove ITEM_NUM as well.

7. Click Next.

   The wizard displays the Choose Components page.

8. Choose `JFC JTable` from the Component arrangement drop-down list.



9 Check the row numbers checkbox and click Next.

The wizard displays the Master Detail Join Definition page.

10 Select the alias for the `ORDERS` table's `QueryNavigator` in the drop-down list.

11 Define the master-detail relationship as follows:

```
order_num     =   order_num
```

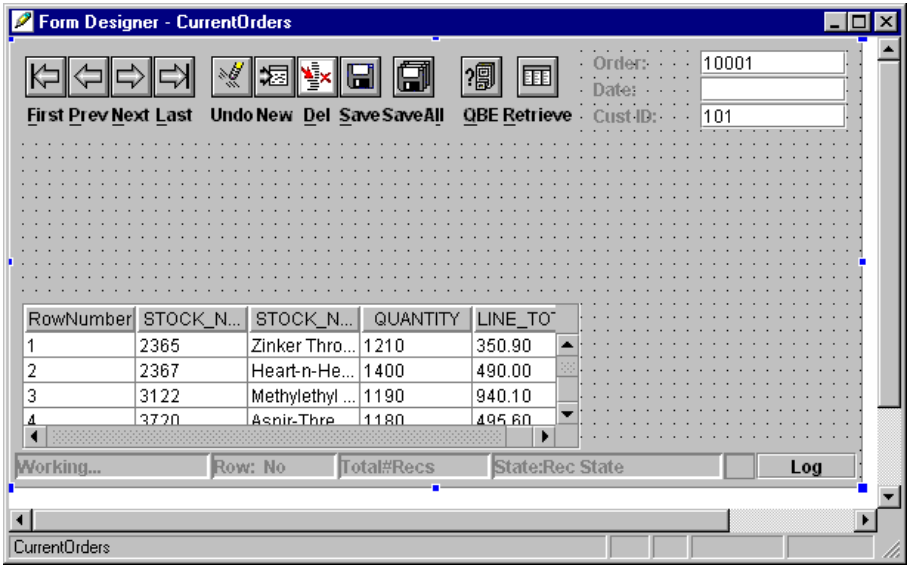
12 Click Next.

13 On the Layout Options page, un-check Statusbar and Toolbar and click Next.

The wizard displays the Summary page that shows what you have selected.

14 Click Finish to add the table.

15 Rearrange the components as shown below and save your changes.



Notice that the databound components show live data from the database.

Adding a detail table with dbNAVIGATOR

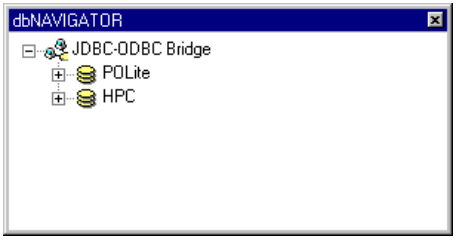
The dbNAVIGATOR enables you to view data sources and the objects (tables, columns, stored procedures) contained in them. It also allows you to drop components onto the form that are bound to these data objects.

Now you will use the dbNAVIGATOR to add the `CUSTOMER` table to the form.

To display the CUSTOMER table in the dbNAVIGATOR window:

1. Choose dbNAVIGATOR from the View menu to display dbNAVIGATOR.
2. Click the plus sign (+) next to the server labeled `JDBC-ODBC Bridge` in the dbNAVIGATOR window.

   dbNAVIGATOR expands the list to display the data sources available.

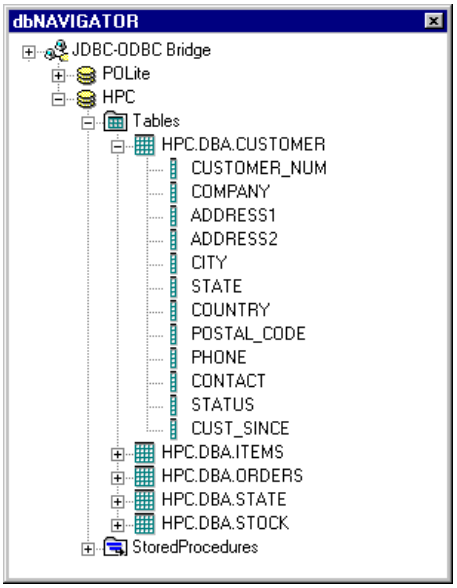


3. Now expand the `HPC` data source and its `Tables`.

   If this is the first time you have connected to the database during this session, VisualCafé displays the User Authentication dialog box

4. Enter `DBA` in the User name field and click OK (no password is necessary).

5 Expand the `CUSTOMER` table to display its columns.



To add the CUSTOMER table:

1 CTRL-click to select all of the columns *except* `CUSTOMER_NUM`, `ADDRESS2`, and `CUST_SINCE`.

2 Drag the columns from the dbNAVIGATOR window and drop them in an open space on the `CurrentOrders` form.

VisualCafé generates the following:

- the connection objects (a `QueryNavigator` and a `RecordDefinition`) for the table
- a `JTextField` and `JLabel` for each column
- a `BindingModel` object to bind each field to its respective column

For more information about these objects, refer to the *VisualCafé User's Guide: Expert Features*.

3 In the Project window, click the QueryNavigator icon for the `CUSTOMER` table.

4 In the Property List, double-click the Join Columns field.

VisualCafé displays the Master Detail Join Definition box.

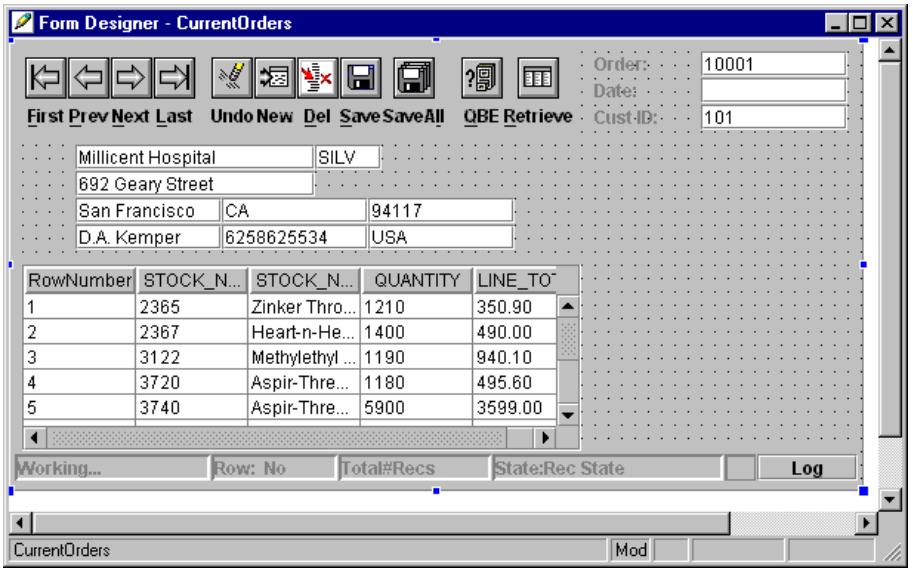
5 Select the alias for the `ORDERS` table in the drop-down list.

6 Define the master-detail relationship as follows:

```
customer_num     =   customer_num
```

7 Click Finish.

8 Delete the auto-generated `JLabel`s and arrange the components as shown below.



9 Save your changes.

The text that will appear in the `PHONE` field is simply a 10-digit number. To make this text more readable, replace the `JTextField` with a `JMaskedTextField`.

To add the JMaskedTextField:

1 Delete the `PHONE JTextField`.

2 Click the `JMaskedTextField` component (on the Swing Additions tab of the Component Palette), drag it to the Form Designer, and drop it on the applet.

3 In the Property List, enter `PHONE` for the Name value.

4 Reposition the `JMaskedTextField` to where the `JTextField` was.

5 Click on the `PHONEBindingModel` component in the Project window.

6. In the Property List, select PHONE in the Component property's drop-down list.
7. Save your changes.

If you'd like to see how it works so far, choose Execute from the Project menu to run the applet. When you are finished close the applet by choosing Quit from its Applet menu.

Adding a lookup column

Now you can add a column to display the full name of the state instead of just the two-letter abbreviation. To do this you will need to add the `STATE` table to the form, create a `JComboBox`, and then create the lookup binding.

To add the STATE table:

1. If necessary, choose dbNAVIGATOR from the View menu to display dbNAVIGATOR.
2. Click the plus sign (+) next to the server labeled `JDBC-ODBC Bridge` in the dbNAVIGATOR window.
3. Now expand the `HPC` data source and its `Tables`.

   If this is the first time you have connected to the database during this session, VisualCafé displays the User Authentication dialog box
4. Enter `DBA` in the User name field and click OK (no password is necessary).
5. Expand the `STATE` table to display its columns.
6. Drag the `STATE_CODE` and STATE_NAME columns from the dbNAVIGATOR and drop them onto the form.

   VisualCafé generates a QueryNavigator and a RecordDefinition for the table. It also generates a `JTextField`, `JLabel`, and `BindingModel` for each column, which you will not be using.
7. Delete the new `JTextField`s, `JLabel`s, and `BindingModel`s.

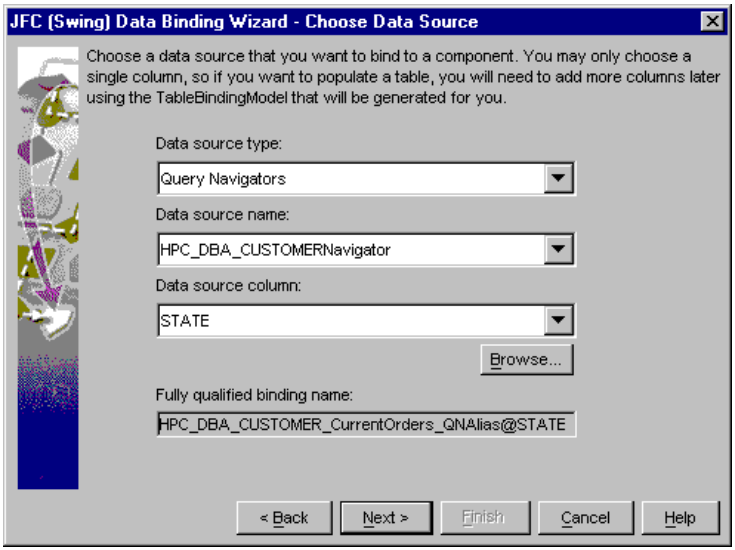
To add and bind the JComboBox:

1. Delete the `STATE JTextField`.
2. Click on the applet in the Form Designer.
3. Choose Add/DataBind Component from the Database menu.

VisualCafé launches the Data Binding Wizard.

4 Click Next on the Introduction page to continue.

The wizard displays the Choose Data Source page.



5 Select Query Navigators from the Data source type drop-down list.

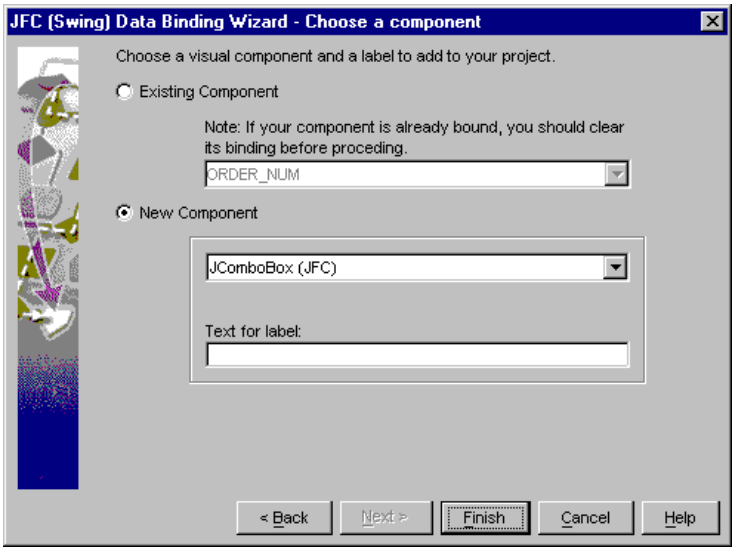
6 Select the name of the `CUSTOMER` table's QueryNavigator from the Data source name drop-down list.

7 Select STATE from the Data source column drop-down list.

8 Click Next to display the Choose a component page.

9 Click the New Component radio button.

10 Select JComboBox from the drop-down list.



11 Delete the text for the label (so VisualCafé doesn't generate a label) and click Finish.

VisualCafé generates a `JComboBox` and a `BindingModel` for it.

12 Reposition it to where the `JTextField` was.

To customize the lookup binding:

1 Click on the new `BindingModel` in the Project window.

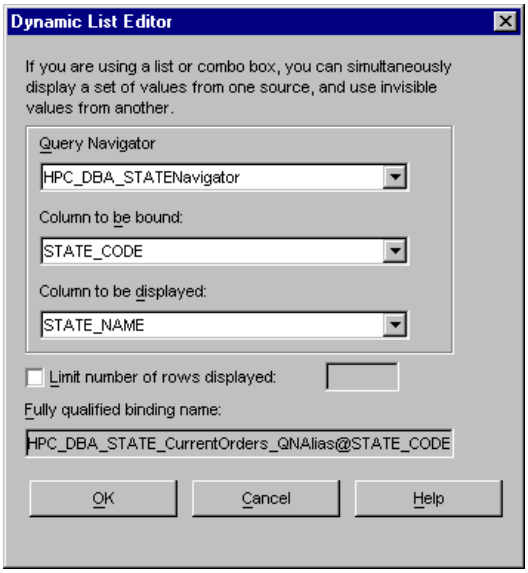
2 Double-click on the Dynamic List field in the JFC Data Binding group of the Property List.

VisualCafé displays the Dynamic List Editor.

3 Select the name of the STATE table's QueryNavigator in the Query Navigator drop-down list.

4 Select STATE_CODE in the Column to be bound drop-down list.

5 Select STATE_NAME in the Column to be displayed drop-down list.



6 Click OK.

7 Click on the Project window and choose Save All from the File menu to save your work.

Adding calculations and validations

With VisualCafé's `CalculationAdapter` and `ValidationAdapter`, you can quickly and easily create rules to perform calculations and validate entries. In this section, you will create calculations for the following:

- SubTotal — to add the values in the `LINE_TOTAL` column of the `ORDERS` table.
- Tax — to determine the sales tax on the order.
- Total — to add the SubTotal and Tax values.

You will also create a validation to notify the user when the value of the Total field exceeds $4,000.

Adding calculated fields

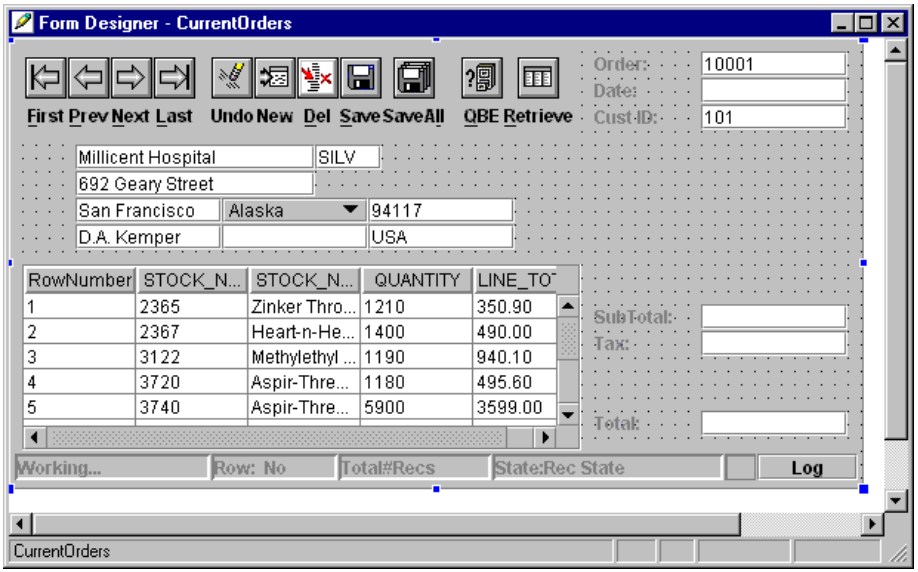
For the `CurrentOrders` applet you will use a `CalculationAdapter` to calculate the SubTotal, Tax, Total. First, you must place the fields on the form.

To put the fields on the form:

1. Drop three `JLabel`s from the Swing tab of the Component Palette onto the form.
2. Drop three `JCurrencyTextField`s from the Swing Additions tab onto the form.
3. Adjust their properties as follows:

Component	Name	Text	Width	Height
JLabel		SubTotal:	72	17
JCurrencyTextField	SUBTOTAL		99	17
JLabel		Tax:	72	17
JCurrencyTextField	TAX		99	17
JLabel		Total:	72	17
JCurrencyTextField	TOTAL		99	17

4 Position the new components as shown below.



Now that you have the fields for the calculated values, you will need to add the `CalculationAdapter`s (to perform the calculations) and then the `BindingModel`s (to bind the calculation output to the field).

To add the calculation for the SUBTOTAL field:

1 Click on the applet in the Form Designer.

2 Choose Add Calculated Data from the Database menu.

VisualCafé displays the Introduction page of the Calculation Customizer Wizard.

3 Click Next.

4 On the Choose Calculation page, click the plus sign (+) next to Calculation Formulas to display the available formulas.

5 Select AdditionFunction and click Next.

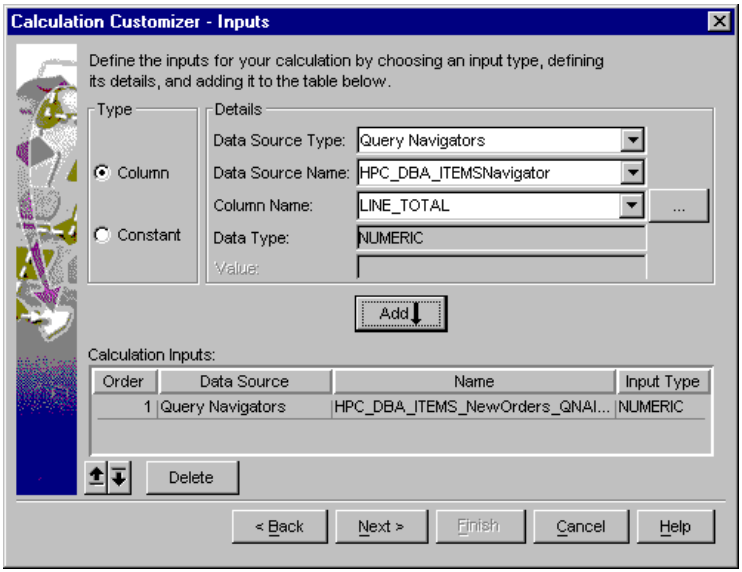
The wizard displays the Inputs page.

6 Click the Column radio button and select the following options:

Data source type = `QueryNavigators`

Data source name = the name of the ITEMS table's `QueryNavigator`

Column name = `LINE_TOTAL`



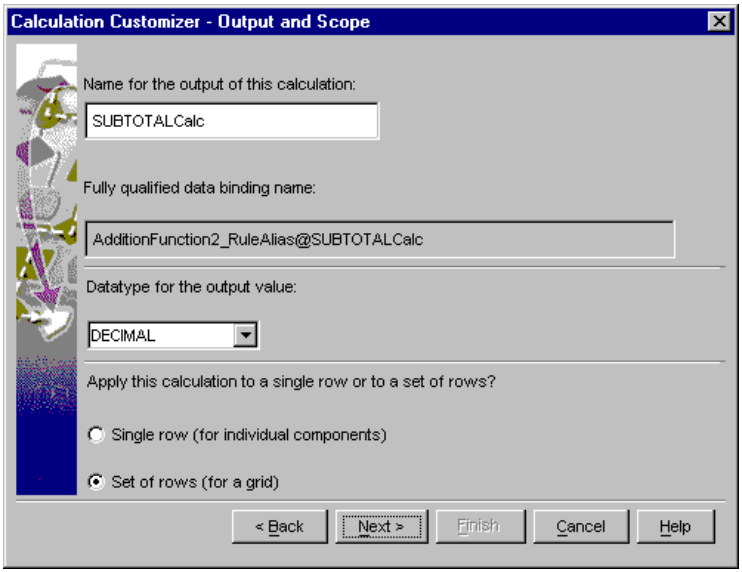
7 Click Add, then click Next.

8 On the Output and Scope page, enter `SUBTOTALCalc` in the Name field.

Notice that the fully qualified name expands as you type.

9 Select DECIMAL from the Datatype drop-down list.

10 Choose Set of rows, and click Next.



The wizard displays a summary page to show you what you have selected.

11 Click Finish to generate the code for the calculation.

To add the BindingModel for the SUBTOTAL field:

1 Drop a `BindingModel` from the Swing DataBound tab of the Component Palette on the form.

2 In the Property List, select SUBTOTAL in the drop-down list of the Component property in the JFC Data Binding group.

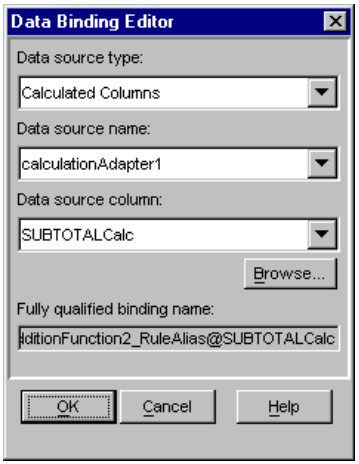
3 Double-click in the Data Binding field of the Property List.

VisualCafé displays the Data Binding Editor.

4 Select Calculated Columns from the Data source type drop-down list.

5 Select the name of the `CalculationAdapter` from the Data source name drop-down list.

6 Select SUBTOTALCalc from the Data source column drop-down list.



7 Click OK.

To add the calculation for the TAX field:

1 Click on the applet in the Form Designer.

2 Choose Add Calculated Data from the Database menu.

VisualCafé displays the Introduction page of the Calculation Customizer Wizard.

3 Click Next.

4 On the Choose Calculation page, expand the Calculation Formulas folder and select the MultiplicationFunction and click Next.

5 On the Inputs page, click the Column radio button and select the following options:

Data source type = `Calculated Columns`

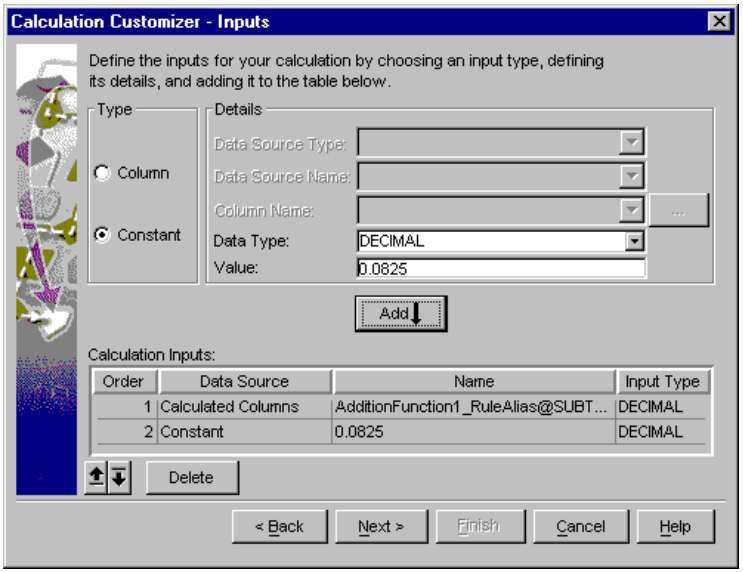
Data source name = the name of the `CalculationAdapter` for `SUBTOTAL`

Column name = `SUBTOTALCalc`

6 Click Add.

7 Then click the Constant radio button and select DECIMAL from the Data Type drop-down list.

8 Enter `0.0825` in the Value field and click Add.



9 Click Next.

10 On the Output and Scope page, enter `TAXCalc` in the Name field.

11 Select DECIMAL from the Datatype drop-down list.

12 Choose Single Row, and click Next.

The wizard displays the summary page.

13 Click Finish to generate the code for the calculation.

To bind the calculation for the TAX field:

1 Drop a `BindingModel` from the Swing DataBound tab of the Component Palette onto the form.

2 In the Property List, select TAX in the drop-down list of the Component property in the JFC Data Binding group

3 Double-click in the Data Binding field of the Property List.

VisualCafé displays the Data Binding Editor.

4 Select Calculated Columns from the Data source type drop-down list.

5 Select the name of the new `CalculationAdapter` from the Data source name drop-down list.

6 Select TAXCalc from the Data source column drop-down list.

7 Click OK.

To add the calculation for the TOTAL field:

1. Click on the applet in the Form Designer.
2. Choose Add Calculated Data from the Database menu.

   VisualCafé displays the Introduction page of the Calculation Customizer Wizard.
3. Click Next.
4. On the Choose Calculation page, expand the Calculation Formulas folder and select the AdditionFunction and click Next.
5. On the Inputs page, click the Column radio button and select the following options:

   Data source type = `Calculated Columns`

   Data source name = the name of the CalculationAdapter for SUBTOTAL

   Column name = `SUBTOTALCalc`
6. Click Add.
7. Then select the following options:

   Data source type = `Calculated Columns`

   Data source name = the name of the `CalculationAdapter` for `TAX`

   Column name = `TAXCalc`
8. Click Add, then click Next.
9. On the Output and Scope page, enter `TOTALCalc` in the Name field.
10. Select DECIMAL from the Datatype drop-down list.
11. Choose Single Row, and click Next.

    The wizard displays the summary page.
12. Click Finish to generate the code for the calculation.

To bind the calculation for the TAX field:

1. Drop a `BindingModel` from the Swing DataBound tab of the Component Palette on the form.
2. In the Property List, select TOTAL in the drop-down list of the Component property in the JFC Data Binding group.
3. Double-click in the Data Binding field to display the Data Binding Editor.
4. Select Calculated Columns from the Data source type drop-down list.

5. Select the name of the new `CalculationAdapter` from the Data source name drop-down list.
6. Select TOTALCalc from the Data source column drop-down list.
7. Click OK.
8. Click on the Project window and choose Save All from the File menu to save your work.

Adding a validation

At Hartwig, orders that exceed $4,000 must be approved by a manager. To notify a manager using this applet that an order needs to be approved, you can use a validation rule. Here you'll add a validation that checks the total and displays a message box when the total exceeds the limit.

To add the validation:

1. Click on the applet in the Form Designer.
2. Choose Add Validation from the Database menu.

   VisualCafé displays the Introduction page of the Validation Customizer Wizard.
3. Click Next.
4. On the Choose Calculation page, click the plus sign (+) next to Validation Formulas to display the available formulas.
5. Select NumberValidation and click Next.

   The wizard displays the Inputs and Operators page.
6. In the This value group, select the following:

   Data source Type = `Calculated columns`

   Data source Name = the name of the `TOTAL CalculationAdapter`

   Input = `TOTALCalc`

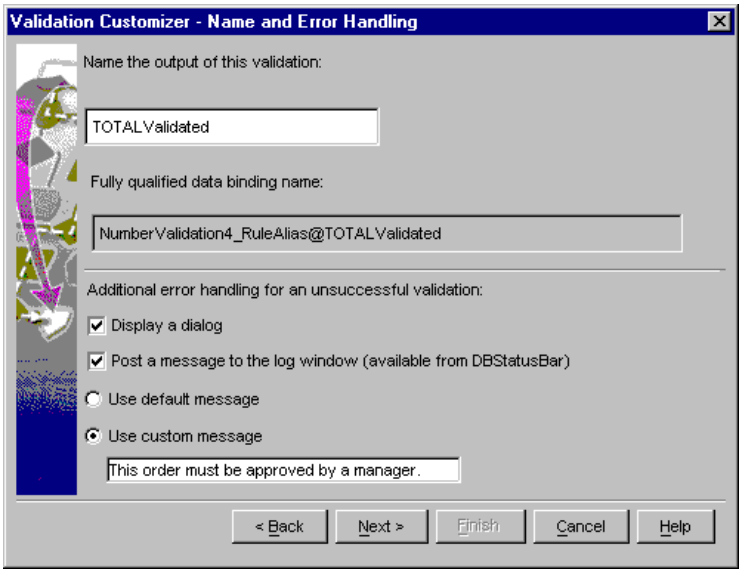
   The Data Type should be automatically set to `DECIMAL`.
7. Select the `<` (less than) operator.
8. In the Compared to This value group, click the Constant radio button and enter `4000` in the corresponding field.
9. Click Next.

   The wizard displays the Name and Error Handling page.
10. Enter `TOTALValidated` in the Name field.

11 Make sure both the Display a dialog and Post a message to the log window checkboxes are checked.

12 Click the Use custom message radio button and enter the following in the corresponding field.

```
This order must be approved by a manager.
```



13 Click Next to display the Summary page.

14 Click Finish to generate the validation rule.

15 Click on the Project window and choose Save All from the File menu to save your work.

Cleaning up the user interface

Now there are just a few details to clean up. You need to do the following:

- Provide masks for the `JMaskedTextField`s.
- Make all the fields except the order number and customer number uneditable.
- Customize the `ITEMS` table.

To provide masks for the JMaskedTextFields:

1. Click on the RECEIVE_DATE field in the Form Designer.
2. In the Mask field of the Property List, enter `9999999999` (10 nines).
3. Click on the PHONE field in the Form Designer.
4. In the Mask field of the Property List, enter `999-999-9999`.

To make the fields uneditable:

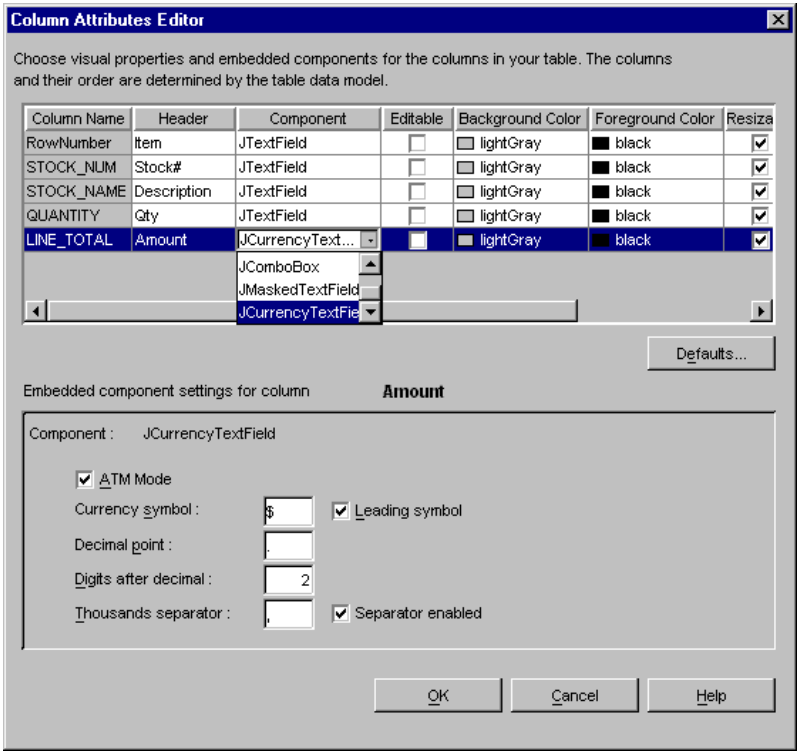
1. Shift-click to select all `JCurrencyTextField`s, all `JMaskedTextField`s, and all `JTextField`s except `ORDER_NUM` and `CUSTOMER_NUM`.
2. In the Property List, select false from the Editable drop-down list.
3. Then select false from the Opaque drop-down list.

To customize the JTable:

1. In the Project window, click on the `TableBindingModel` for the `ITEMS` table.
2. In the Property List, double-click on the Column Attributes field.

   VisualCafé displays the Column Attributes Editor. This editor allows you to modify display attributes for a databound `JTable`.
3. Make all of the columns non-editable by un-checking the Editable checkbox for each column.
4. Change the Background Color for all fields to lightGray.

5 Change the component for the `LINE_TOTAL` column by selecting JCurrencyTextField from the Component drop-down list.



6 Change the Header and Width attributes as follows:

Column	Header	Width
`RowNumber`	`Item`	`40`
`STOCK_NUM`	`Stock#`	`40`
`STOCK_NAME`	`Description`	`170`
`QUANTITY`	`Qty`	`60`
`LINE_TOTAL`	`Amount`	`60`

7 Click OK to apply the changes.

What's Next?

Congratulations! You've completed the tour. The tasks you have completed along the way were just a small sample of what you can do with VisualCafé. Now you're ready to develop your own Java applets and applications. For more assistance as you continue, explore the online help, the *VisualCafé User's Guide: Standard Features*, and if applicable the *VisualCafé User's Guide: Expert Features*. Enjoy your journey.